DOCUMENT RESUME

ED 339 882 CE 059 831

AUTHOR Casey, Joe
TITLE CNC Programming I.
INSTITUTION Milwaukee Area Technical Coll., Wis.
SPONS AGENCY Wisconsin State Board of Vocational, Technical, and Adult Education, Madison.
PUB DATE 90
NOTE 82p.; For a related manual, see CE 059 832.
PUB TYPE Guides - Classroom Use - Teaching Guides (For Teacher) (052)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS Behavioral Objectives; Competence; Competency Based Education; *Computer Assisted Manufacturing; *Computer Oriented Programs; Computer Software; *Computer Software Development; *Course Content; Electronic Control; Learning Activities; Learning Modules; Machine Tools; *Numerical Control; Programing; Teaching Methods; *Technical Education; Technical Institutes; Two Year Colleges; Units of Study
IDENTIFIERS Cartesian Coordinates

ABSTRACT

This document contains five units for a course in computer numerical control (CNC) for computer-aided manufacturing. It is intended to familiarize students with the principles and techniques necessary to create proper CNC programs manually. Each unit consists of an introduction, instructional objectives, learning materials, learning activities, and assignments. The units cover the following topics: Cartesian coordinates, absolute and incremental positioning, basic manual programming, polar coordinates and cutter compensation, and programming the Heidenhain controller. An appendix contains blank forms for CNC tool lists and operations and for numerical control programs. (KC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

09-473
WI TO CD 12

CNC Programming 1

by Joe Casey

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

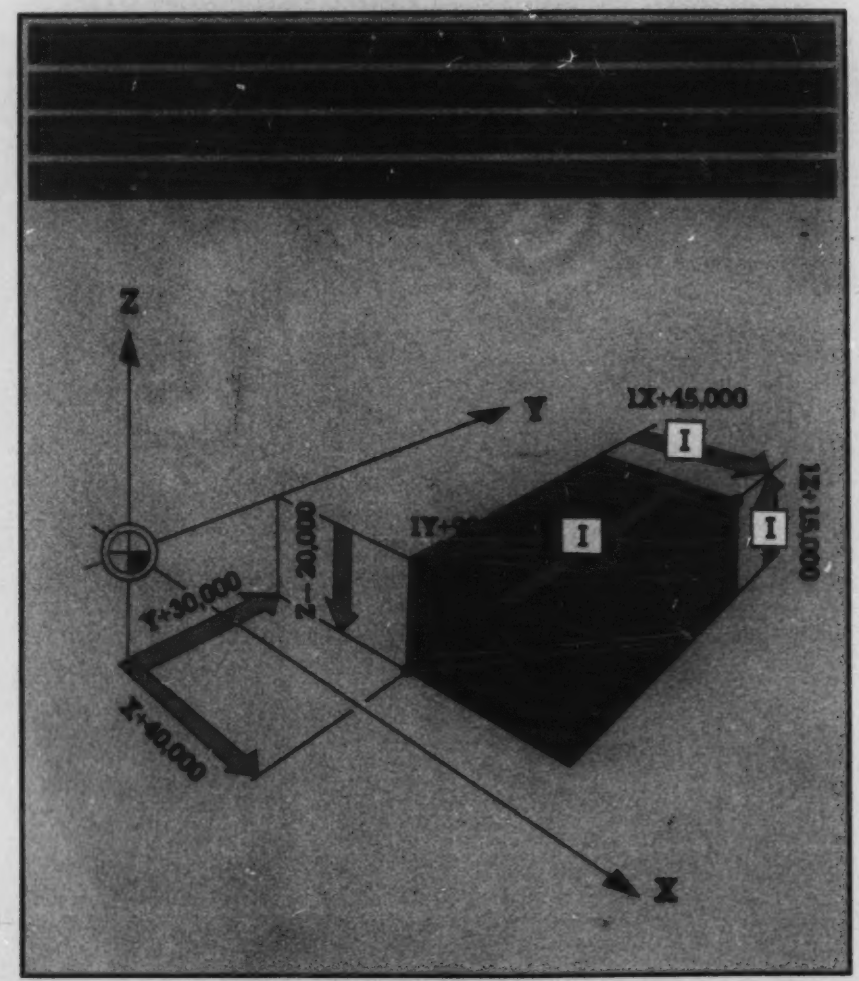
✓ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED339882



matc

Milwaukee Area Technical College

CE 059 831

BEST COPY AVAILABLE

The author would like to thank the following people for their help. You know who you are and what you did. Carlson Tool, Randy at Carlson Tool, Wayne B., Joyce E., Vicky B., Bowen P., Roberta S., and Judy S. Thank you all.

MILWAUKEE AREA TECHNICAL COLLEGE
Syllabus
CNC Programming I
TD MKG 439-365

1.	Instructor	Joe Casey Office T211 Phone 278-6532
2.	Office Hours	_____
3.	Class Meets	_____
4.	Credits	1
5.	Periods	2
6.	Prerequisites	439-365 students must be currently enrolled in the 439 shop classes.
7.	Textbooks and Supplies	CNC Programming I, Information and Assignment Units, (MATC) by Casey, 1 pad of graph paper, a calculator
8.	Course Description	The purpose of this course is to familiarize the student with the principles and techniques necessary to manually create proper CNC programs.
9.	Course Outline	

Session 1	Date_____	Introduction
Session 2	Date_____	Chapter 1
Session 3	Date_____	Chapter 1
Session 4	Date_____	Chapter 2
Session 5	Date_____	Chapter 2
Session 6	Date_____	Chapter 3
Session 7	Date_____	Chapter 3
Session 8	Date_____	Chapter 3
Session 9	Date_____	MIDTERM EXAM
Session 10	Date_____	Chapter 4
Session 11	Date_____	Chapter 4
Session 12	Date_____	Chapter 4
Session 13	Date_____	Chapter 5
Session 14	Date_____	Chapter 5
Session 15	Date_____	Chapter 5
Session 16	Date_____	FINAL EXAM

10.	Assignments	Assignments for each unit will be given in class at least 1 week in advance.
11.	Requirements	Students are required to complete all assignments and tests. Attendance at every session is expected and attendance will be taken.
12.	Grading	Assignments 50%, Each Test 25%.
13.	Attendance	Attendance will be per T&I Divisional policy.
14.	Makeup Procedures	Tests must be made up in advance. 10 points will be deducted from a test score for each day that it is taken late.
15.	Dropping Courses	Per District policy as explained in the Student Handbook.
16.	Academic Alert System	Per District policy as explained in the Student Handbook.
17.	Optional References	Reference materials on CNC Programming may be found in the College Library and in the tool and die shop (T106).
18.	Bibliography	No outside reading is required for this course.

MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

UNIT 1 Cartesian Coordinates

INTRODUCTION

CNC machine tool movement is based on the Cartesian system of defining coordinate points. These points are defined by how far and in what direction they are from the center of the intersection of two lines.

INSTRUCTIONAL OBJECTIVES

When you have completed this unit you will be able to :

1. Correctly reproduce a Cartesian Coordinate Grid.
2. Accurately position points on a Cartesian Grid.
3. Summarize the historical development of CNC.

LEARNING MATERIALS

1. CNC PROGRAMMING I, Information and Assignment Unit 1.

LEARNING ACTIVITIES

1. Read CNC Programming I, Information Unit 1. (Objectives 1-3)
2. Complete CNC Programming I, Assignment Unit 1. (Objectives 1-3)

ASSIGNMENT

1. Read Information Unit 1.
2. Attend the Lecture/Discussions on Unit 1.
3. Complete Assignment Unit 1.

INFORMATION UNIT 1

Welcome to the world of CNC programming. Before we go any further lets define those initials - CNC. CNC stands for Computerized Numerical Control. This means that a computer is used to move a machine tool, such as a milling machine, through a series of programmed moves. These moves are programmed into the computer's memory via a series of coded instructions. These "programmed" and "coded" instructions make up what is called a CNC Program. People who create these programs are called programmers. This course will teach you how to take a part drawing and translate that information into a CNC program which will be used by the computer to tell the machine tool to create that part.

In 1947, a man named John Parsons began experimenting with a system of numbers with which he hoped to control a milling machine which was being used to produce aircraft propellers. This was a very difficult operation due to the complexity of the curves on a propeller. In 1949 he was awarded a contract by the Air Force to produce what was to become the first NC machine. "NC" stands for Numerical Control. These early machines did not use computers and relied only on complicated numerical commands.

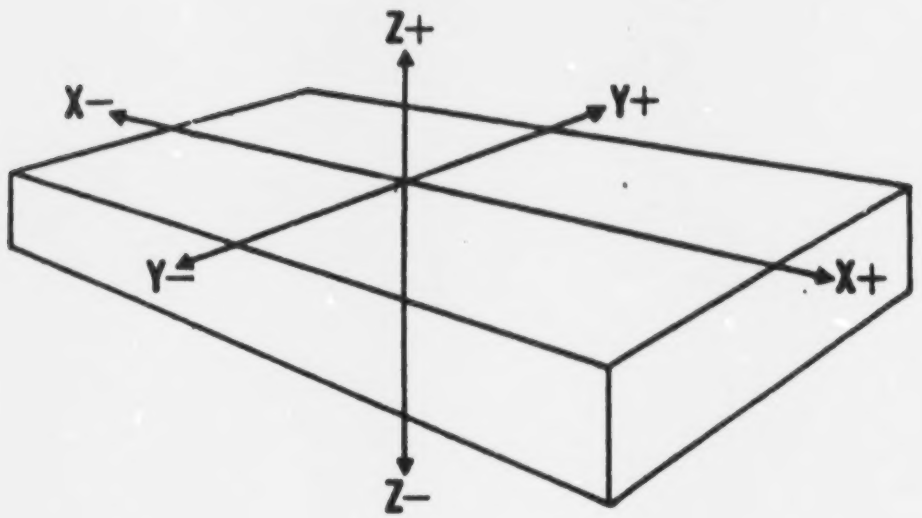
These early machines were called "open looped". This means that there was no direct feedback to the control which verified that the programmed move had actually taken place. If the command was to move 5 inches to the left, there was no way of monitoring the accuracy of the move. This system was not accurate enough for the very close tolerances required in tool and die making. Something better was required.

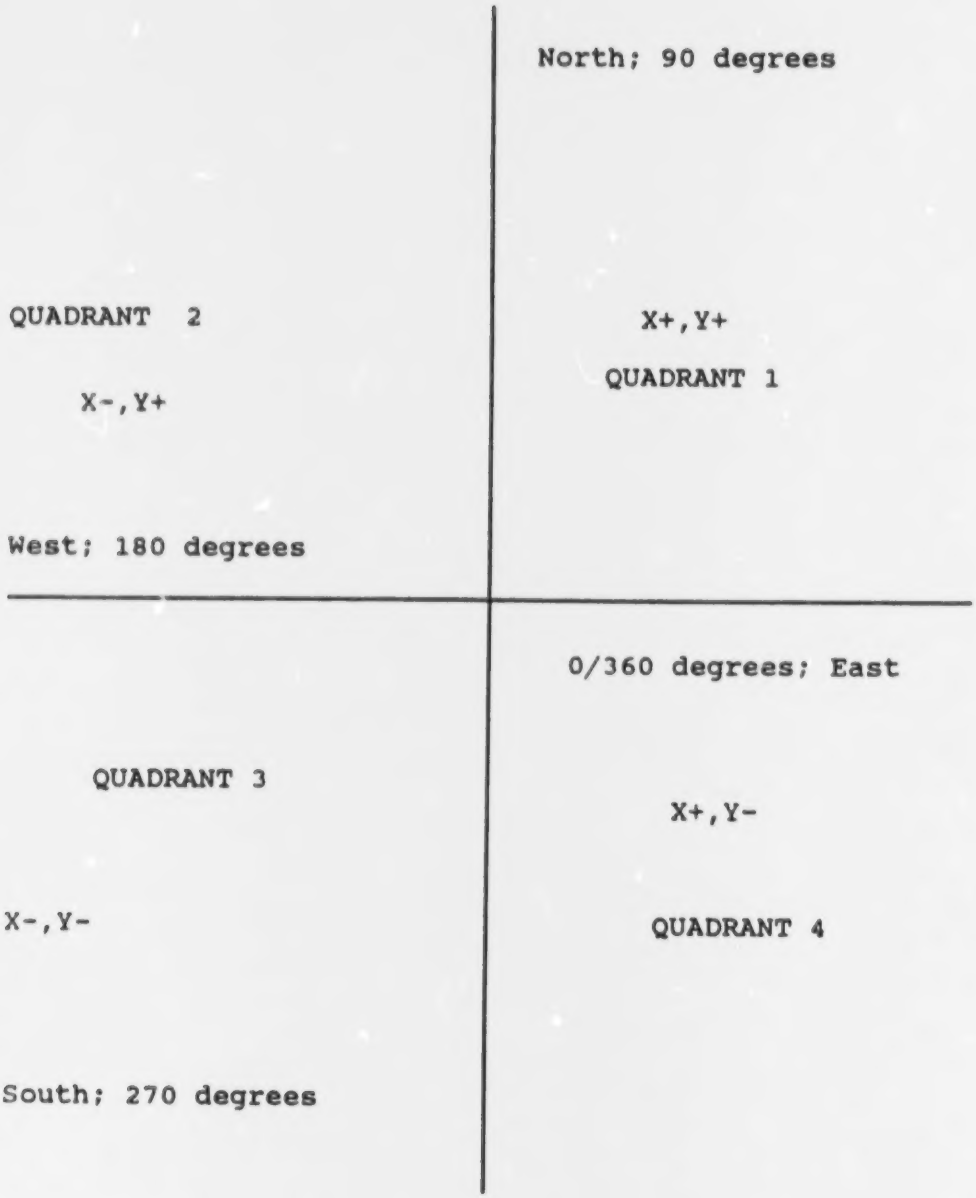
A new system, called the "closed loop" was developed. This method required the use of computers which could very quickly and accurately monitor and adjust to make certain that the required movement had actually taken place. The use of these computers changed NC into CNC which is where we are today.

These programs, once written, can be entered into the machine in a variety of ways. They can be entered directly into the machine by using a system of keyboard entries. They can be stored on paper tape and then transferred electronically into the machine tool via a tape reader. This electronic transfer is called "downloading". They can be stored on cassette tapes and downloaded. They can be created by other computers and then transferred from one computer to another. No matter how they are entered into the machine, they must first be written by a human. Lets begin to learn how this is done.

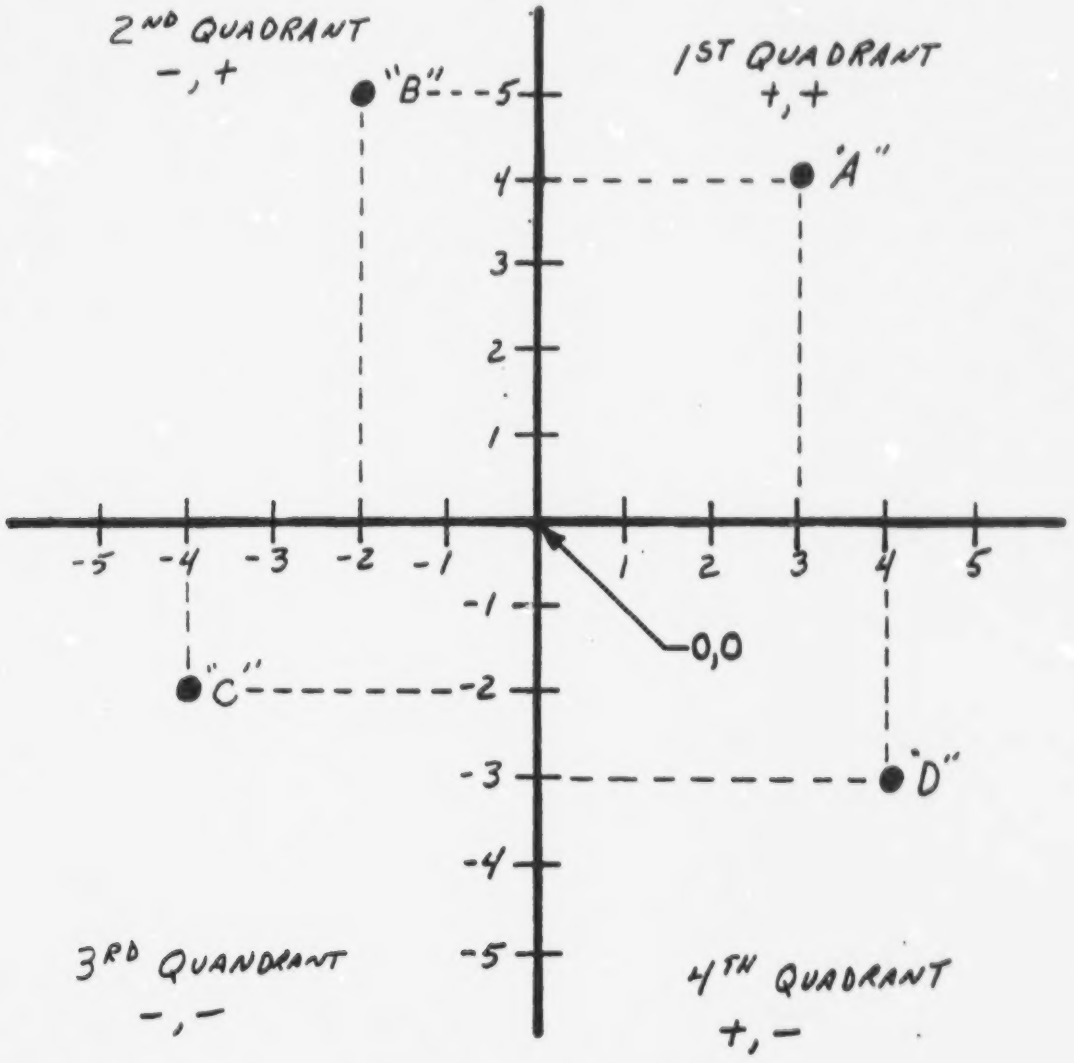
A CNC programmer must have a thorough understanding of the Cartesian Coordinate System. Named for its creator, Rene Descartes - the French philosopher and mathematician -, this system is a way of defining the location point of the intersection of two lines. It lends itself so well to CNC programming that all CNC machines define number locations using this system.

The grid on the next page is an example of this system. The X axis is defined as a horizontal plane (left and right) with X+ being to the right of Zero (NOTE: ZERO IS THE POINT AT WHICH THE TWO LINES INTERSECT ON THE SURFACE OF THE PAGE.) and X- being to the left of Zero. The Y axis is defined as a vertical plane (up and down) with Y+ being above Zero and Y- being below Zero. The Z axis is defined as being perpendicular to the page with Z+ being above the paper and Z- being below the paper.The X, Y,and Z axis are also in these planes on the Bridgeport mill. The following drawing illustrates these axis as they relate to the machine table.

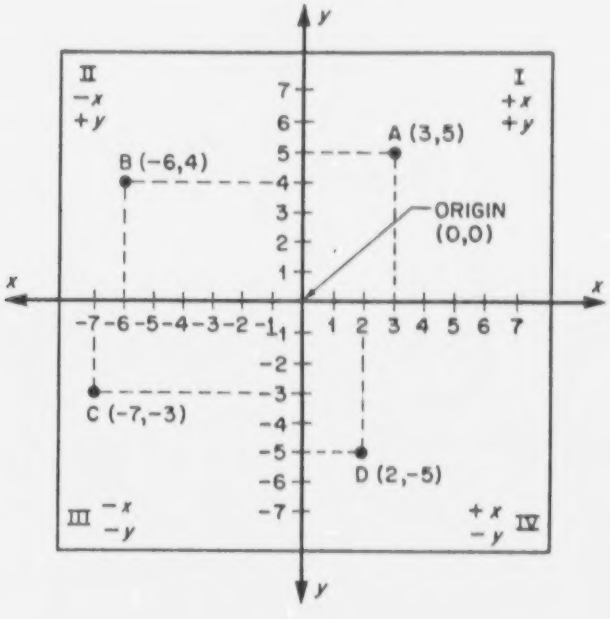



North; 90 degrees
QUADRANT 2
X-,Y+
X+,Y+
QUADRANT 1
West; 180 degrees
0/360 degrees; East
QUADRANT 3
X-,Y-
X+,Y-
QUADRANT 4
South; 270 degrees

Lets look at some location points and see how they would be defined under the Cartesian system. Point A is located in the first Quadrant which means it falls somewhere between 0 and 90 degrees. It is 3 units to the right of zero and 4 units above zero. Since both of these are positive directions; the location of point A is X3,Y4. Notice that all numbers are assumed to be positive unless otherwise stated and that the X location is given first. Using this same procedure, we find that point B is located at X-2,Y5. X is negative because it is to the left of Zero. Point C is X-4,Y-2 and point D is X4,Y-3.



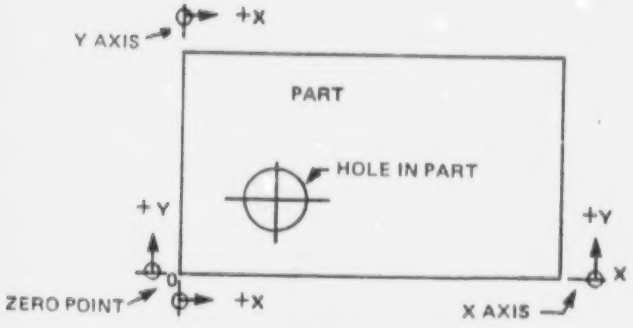
Another example of how the Cartesian Coordinate system works is found in the figure below. Point A has given coordinates of X +3 and Y +5. Since the X coordinate is always given first and since numbers are assumed to be positive unless they have a negative sign (-) in front of them, we can write these coordinates as 3,5. These numbers mean that Point A is 3 units to the right of zero (X+) and 5 units above zero(Y+). All points which have positive X and positive Y coordinates fall within Quadrant 1. Quadrant 1 is between 0 degrees (3 o'clock) and 90 degrees. Point B has coordinates of X-6 and Y4. This means that Point B is 6 units to the left of zero (X-) and 4 units above zero (Y+). All points which have negative X and positive Y coordinates fall within Quadrant 2. Quadrant 2 lies between 90 degrees and 180 degrees. Similarly, Point C has coordinates of X-7, and Y-3. This means that Point C is 7 units to the left of zero (X-) and 3 units below zero (Y-). All point which have coordinates of negative X and negative Y fall within Quadrant 3. Quadrant 3 is between 180 degrees and 270 degrees. Point D has given coordinates of X+2 and Y-5. This means that Point D lies 2 units to the right of zero (X+) and 5 units below zero (Y-). All points which have coordinates of positive X and negative Y fall within Quadrant 4. Quadrant 4 lies between 270 degrees and 360/0 degrees (3 o'clock).



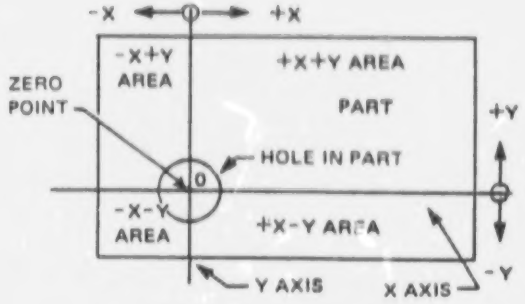
By using the Cartesian Coordinate system to define the location of all points on the part in relation to a point of origin (0,0), by assigning a point of origin to the part, and by referencing the machine tool to that origin point; the programmer is able to tell the machine where to go in relation to a starting point which the machine can understand - the point of origin or part 0,0.

The following 2 figures illustrate how the Cartesian Coordinate system can be applied to a piece part drawing. In Figure A the point of Origin (0,0) is the lower left corner of the piece. This means that the entire piece lies within the First Quadrant (to the right of and above 0,0). This is the most common way of orientating a piece because it allows the programmer to work with positive numbers only. <u>All values are positive within the first Quadrant.</u>

Figure B uses the center of the hole as the point of Origin (0,0). This may be done because the hole is the most critical part of the piece or because a drawing is already dimensioned that way. Because of this point of Origin location, different areas of this piece will fall within all 4 Quadrants. This means that the programmer must use both positive and negative numbers in both X and Y axes.



(A) COORDINATES RUNNING ALONG TWO EDGES OF PART



(B) COORDINATES RUNNING THROUGH PART

MILWAUKEE AREA TECHNICAL COLLEGE

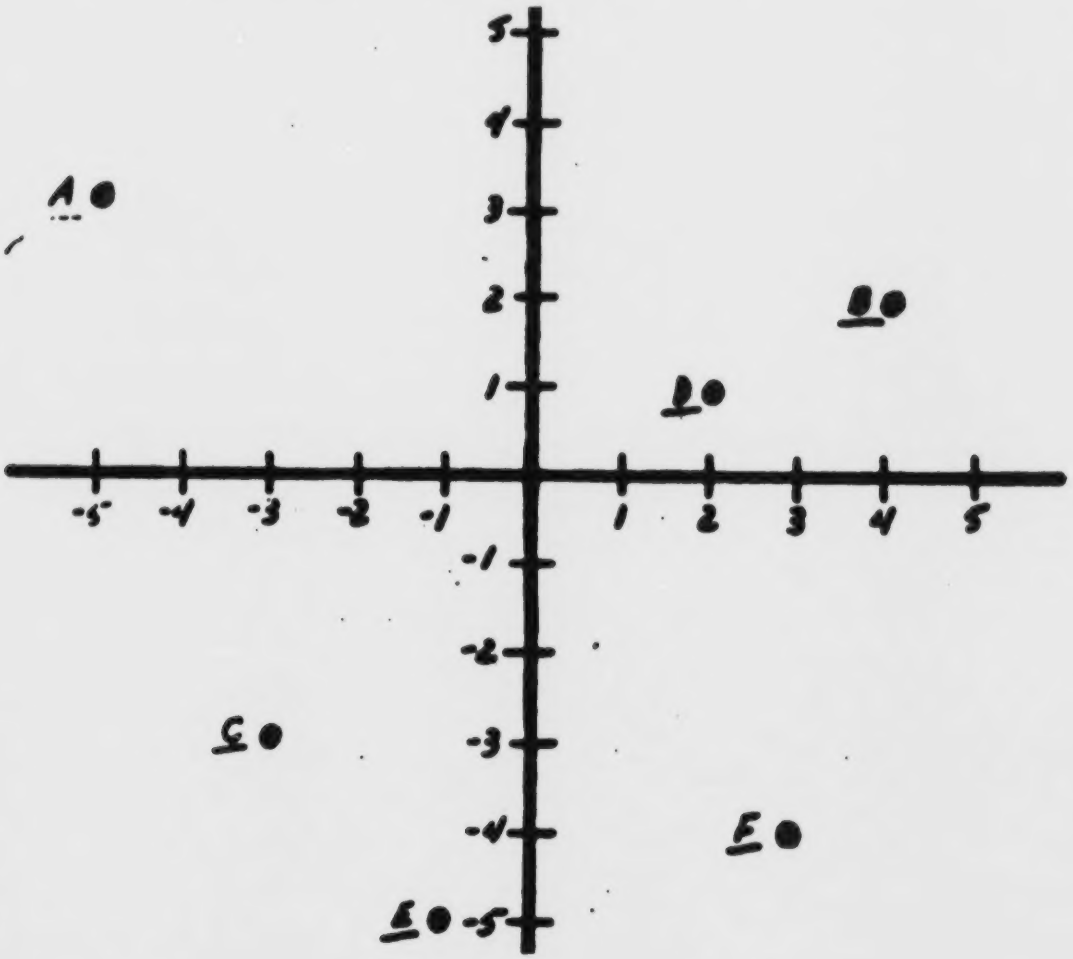
CNC PROGRAMMING I

ASSIGNMENT UNIT 1

1. Identify in which Quadrant the following points lie:

P1 X-3, Y2
P2 X4, Y1
P3 X4, Y-1
P4 X-1, Y-4

2. Give the coordinates for the following points.



3. Using graph paper, plot the following coordinates.

A= -2, 5	B= 3,9	C= -7,-2	D= 0,3
E= -5, 0	F= -2,-2	G= 5,-7	H= -7,5
I= -1,0	J= 0,0	K= -3,-4	L= 8,-6

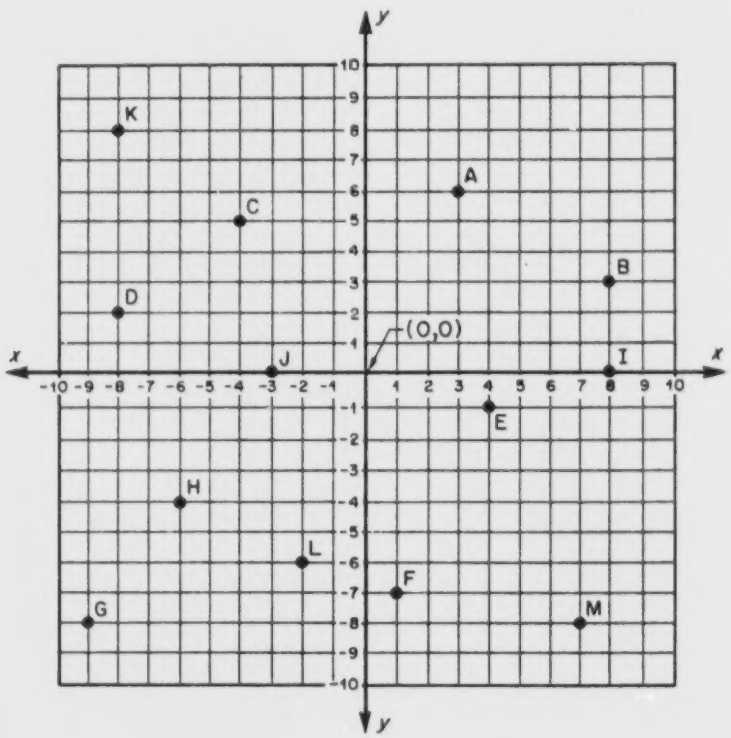
4. Graph and connect the following points: (-5,-5), (-3,-3), (0,0), (2,2), (4,4), and (7,7). What kind of geometric figure is formed?

5. Graph the following points and connect them in the order that they are given.

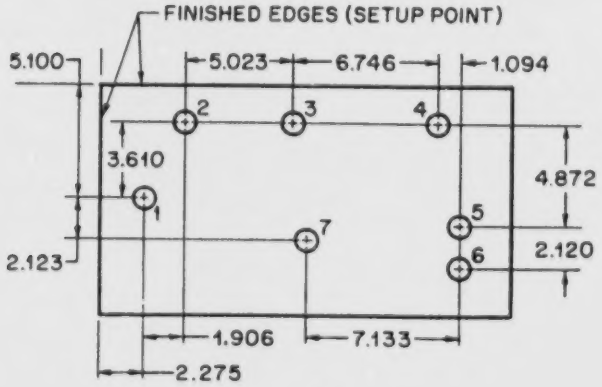
Point 1 (-9,-7), Point 2 (-6,-5.3), Point 3 (-3,-3.5), Point 4 (1,-1), Point 5 (4.5,1), Point 6 (7,2.5) Point 7 (6,3), Point 8 (4,4), Point 9 (2,5), Point 10 (0,6), Point 11 (-2,7), Point 12 (-3,5), Point 13 (-5,1), Point 14 (-6.5,-2), and Point 15 (-8,-5)

What kind of geometric figure is formed?

6. Refer to the points plotted on the figure below. Write the X and Y coordinates for points A through M

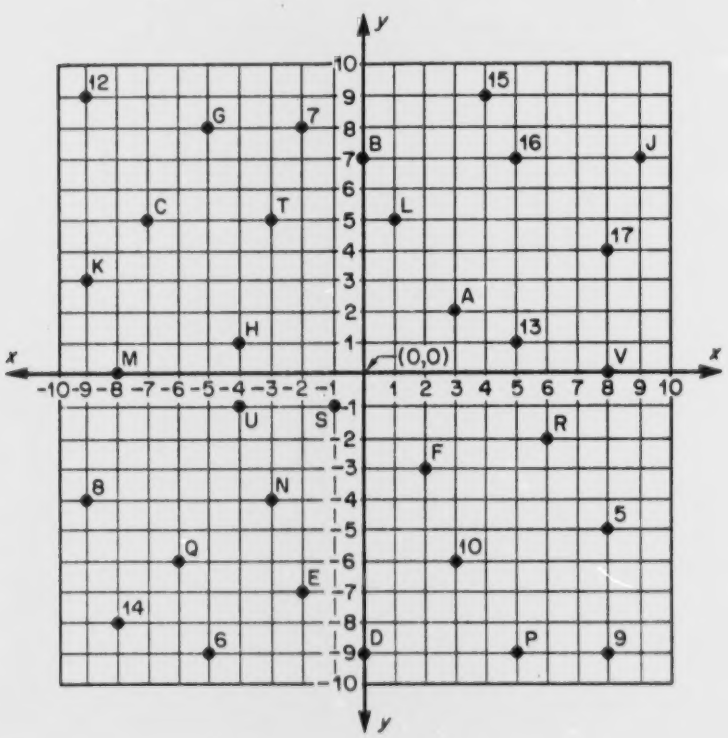


7. Give the X and Y coordinates for the 7 holes on the figure below. Use the upper left corner of the part as 0,0.



10

8. Refer to the points placed on the following figure. Give the X and Y coordinates for letters A-H, J-N, P-V and for numbers 5-17.



BEST COPY AVAILABLE

MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

UNIT 2 Absolute and Incremental Positioning

INTRODUCTION

There are two ways to plot points for a CNC program. One is to relate them all to a common reference point, this is called absolute positioning. The other is to relate them to each other, this is called incremental positioning. In this unit we will examine how coordinate values are assigned in each of these systems.

INSTRUCTIONAL OBJECTIVES

When you have completed this unit you will be able to:

1. Precisely dimension points incrementaly.
2. Precisely dimension points absolutely.
3. Summarize the correct use of G codes as used to define absolute or incremental positioning.

LEARNING MATERIALS

1. CNC Programming I, Information and Assignment Unit 2.

LEARNING ACTIVITIES

1. Read CNC Programming I, Information Unit 2. (Objectives 1-3)
2. Complete CNC Programming I, Assignment Unit 2. (Objectives 1-3)

ASSIGNMENT

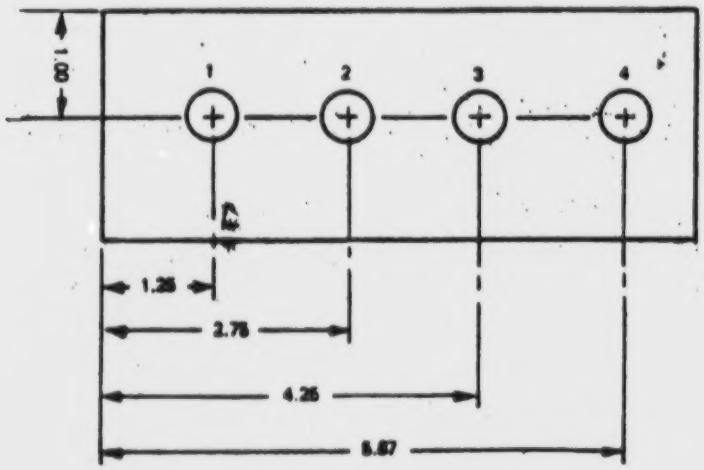
1. Read Information Unit 2.
2. Attend the Lecture/Discussions on Unit 2.
3. Complete Assignment Unit 2.

INFORMATION UNIT 2

In the previous unit we looked at how points can be assigned coordinate values using the Cartesian Coordinate system. In all of our examples we referenced each and every point from a common Point of Origin which we call the 0,0 point. This point is formed by the intersection of the X and the Y axis lines.

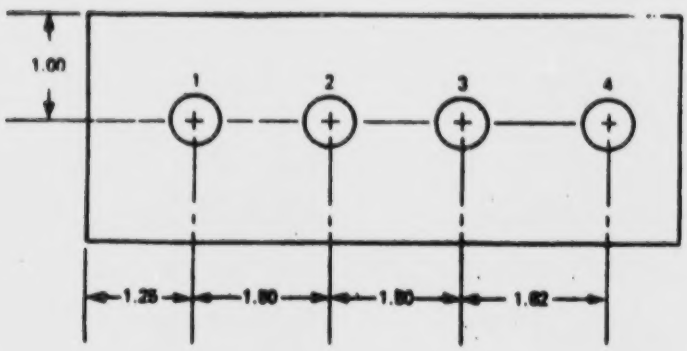
There is a name for this type of referencing. The type of positioning which relates all points to a common point of origin is Absolute Positioning. <u>All points are dimensioned according to how far and in what direction the are from the 0,0 point.</u> This type of positioning is the one that is used most often in CNC programming because it is the easiest and most accurate system. It is the easiest because all dimensions come from the 0,0 point. It is easier to relate things back to one common point than it is to relate them to a series of points. It is the most accurate because all dimensions are taken from a common point of reference which can be established very precisely on the machine tool.

The following figure illustrates a part which has been dimensioned absolutely. All of the holes are dimensioned from and referenced to a common point of origin, the upper left corner of the part. This corner would be called the 0,0 point of the part. Note that since the upper left corner of the piece is 0,0 the entire part fall within the fourth Quadrant. This would require that all X values be positive (to the right of zero) and that all Y values be negative (below zero). These signs will only be used when using the dimensions as part of a CNC program. As dimensions on a drawing they are unsigned.



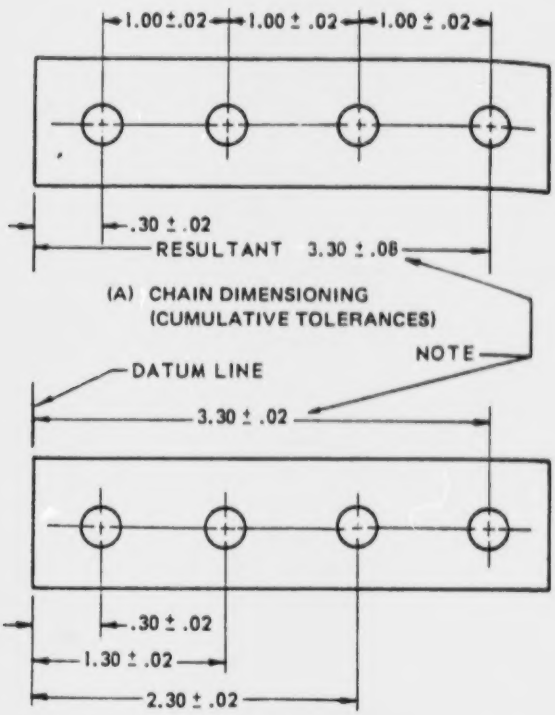
Sometimes, for a variety of reasons, we may wish to locate points in relation to each other. In other words, each point is referenced to the previous point, not to a common 0,0 point. Each dimension is given from the immediate previous location. The location of the previous hole is considered to be 0,0 and the next hole is dimensioned from that previous 0,0 point. Each new location in turn will become 0,0 and the next point will be dimensioned from that 0,0. The direction of travel, X and Y and positive and negative must be noted and is based upon the Cartesian Coordinate system. This type of positioning is called Incremental Positioning.

One of the most common reasons for using this system is that a drawing is already dimensioned that way. If the draftsperson dimensioned each point from the previous point, it is easier to take those dimensions directly from the drawing and transfer them into a CNC program than it is to redo all of the dimensions so that they will relate to a common 0,0 point. The following figure illustrates the same part we looked at earlier. This part is dimensioned incrementally, each hole being dimensioned from the previous one.



One of the problems with incremental dimensioning is that it has the potential to be less accurate that absolute dimensioning due to cumulative tolerances and cumulative error. If each point is referenced to a previous point, then it is only as accurate as that previous point. Since most dimensions will have an applied tolerance, there is the potential for these tolerances to accumulate. This could cause the part to become unacceptable. Figure A illustrates this point. The piece is dimensioned incrementally with the first hole being referenced to the edge of the part. The 1" hole spacing has a + - .02" tolerance applied to it. If the upper limit of that tolerance was used in all of the hole spacings, the total spacing between the first and last hole would be off by .08". This would probably be unacceptable.

Figure B shows the same part dimensioned absolutely. Notice that the worst condition for the first to last hole spacing is .02" range. This is probably within acceptable limits. It is necessary for engineers, draftspersons, and CNC programmers to understand incremental and absolute positioning and to use them correctly.

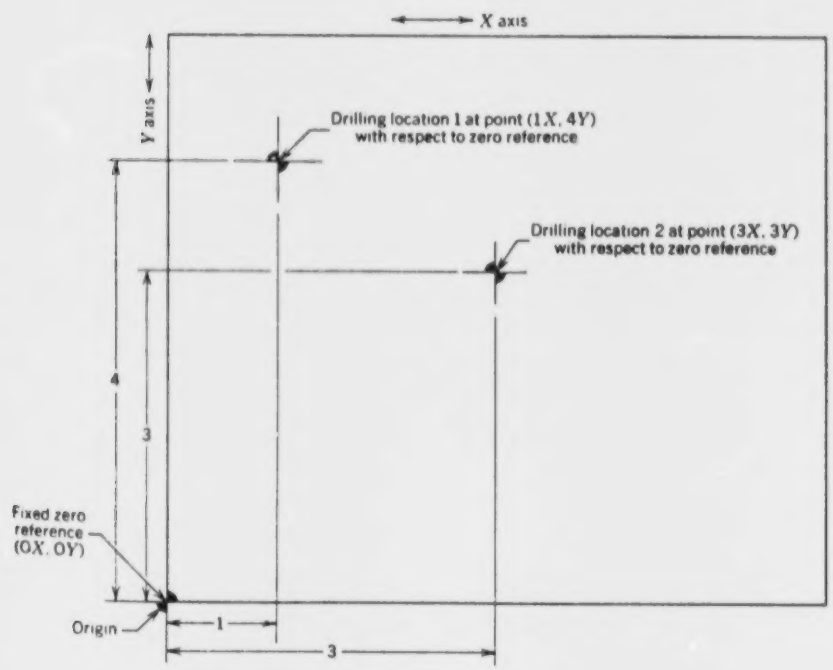


(A) CHAIN DIMENSIONING
(CUMULATIVE TOLERANCES)

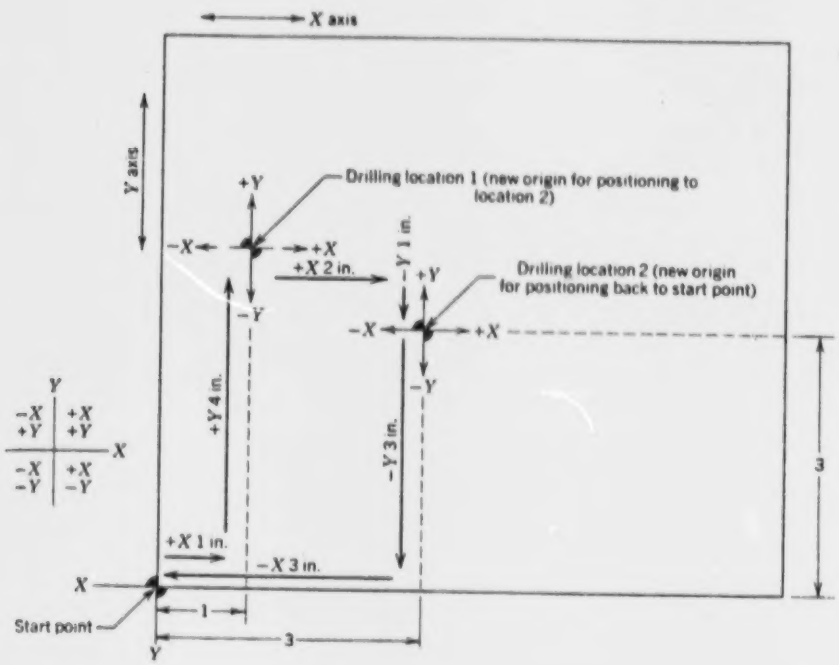
(B) BASE LINE DIMENSIONING
(NON-CUMULATIVE TOLERANCES)

The following figures illustrate how both absolute and incremental positioning could be used in a machining situation. We are going to tell a CNC milling machine to drill 2 holes in a piece of steel. In order to do that we must be able to define for the machine tool where those 2 holes should be located. The first example uses absolute positioning. We tell the CNC machine that we are using absolute positioning via a coded message. These coded messages are called "G" Codes because they all begin with the letter G. There are other codes and we will examine them later. The G code for absolute positioning is G90. Any standard CNC controller will know that a G90 in a program means that all dimensions are referenced to a common 0,0 point. After using the G90 code, we could then tell the machine that the first hole was X+1" and Y+4" from 0,0. 0,0 is the lower left corner of the part. We would move the machine to that corner point and tell it that this point is 0,0 via coded instructions.

We would then tell the machine that the second hole is at X+3 and Y+3 from 0,0.



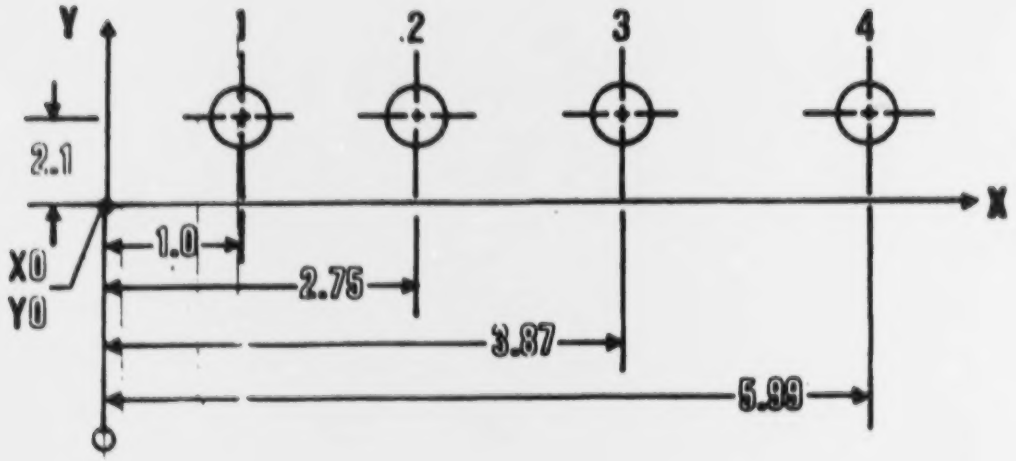
This next figure shows the same part using incremental positioning. The G code for incremental positioning is G91. When this code is entered into the machine controller, it tells the machine to use each previous position as 0,0 and to move the prescribed distance and direction from that point. Since all points must be somehow referenced to the part, notice that the first position in incremental positioning is taken from part 0,0. All other points will then be referenced from the previous point. Using incremental positioning we tell the machine that the first hole is X+1" and Y+4" from 0,0. This is the same as the first dimensions we used in absolute positioning. Now comes the difference! Using incremental positioning, we tell the machine that the second hole is X+2" and Y-1". If we have entered a G91 in our program, the machine will know to move 2" to the right of (X+2) and 1" below (Y-1) hole 1 (0,0) which will put it at the proper location for hole 2.



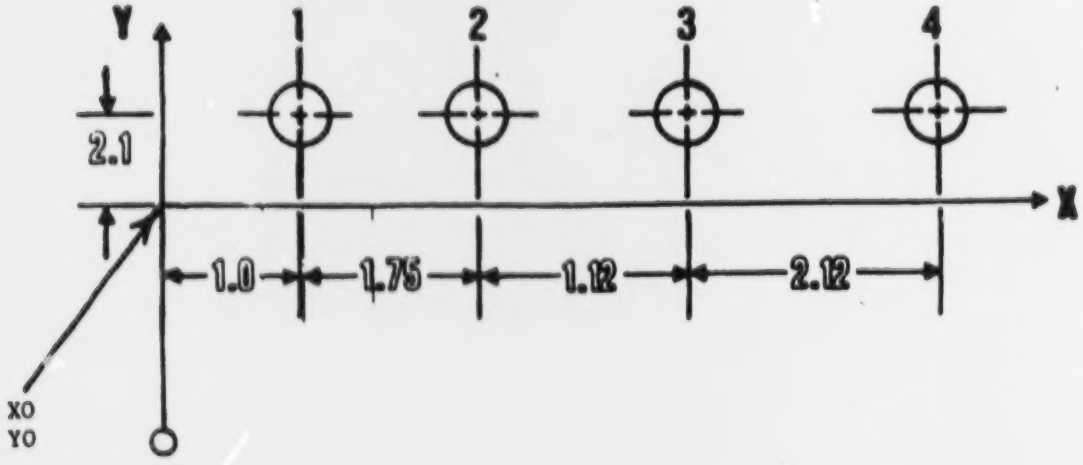
BEST COPY AVAILABLE

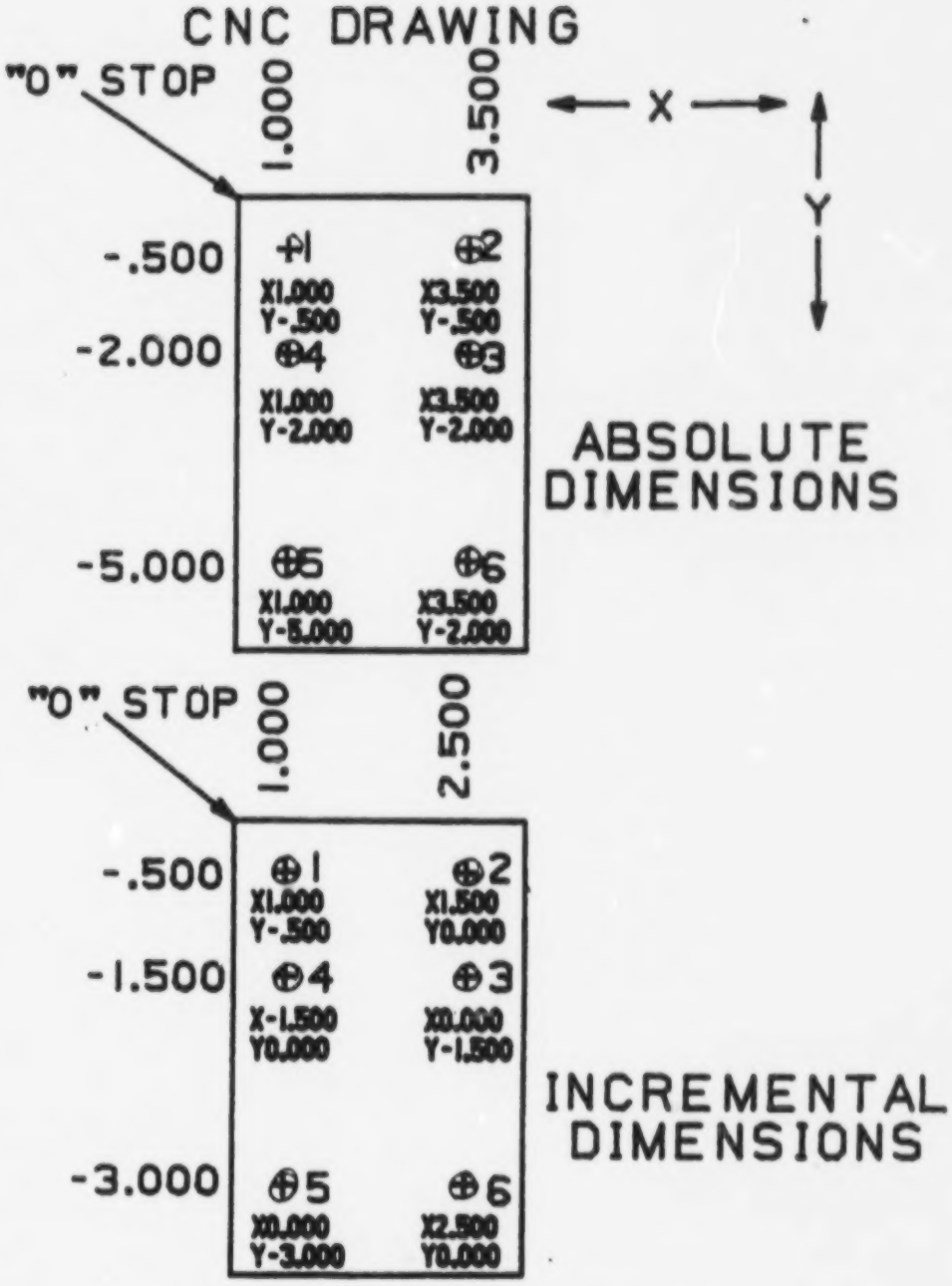
The following figures further illustrate the differences between absolute and incremental positioning.

G90—Absolute Dimensions



G91—Incremental Dimensions



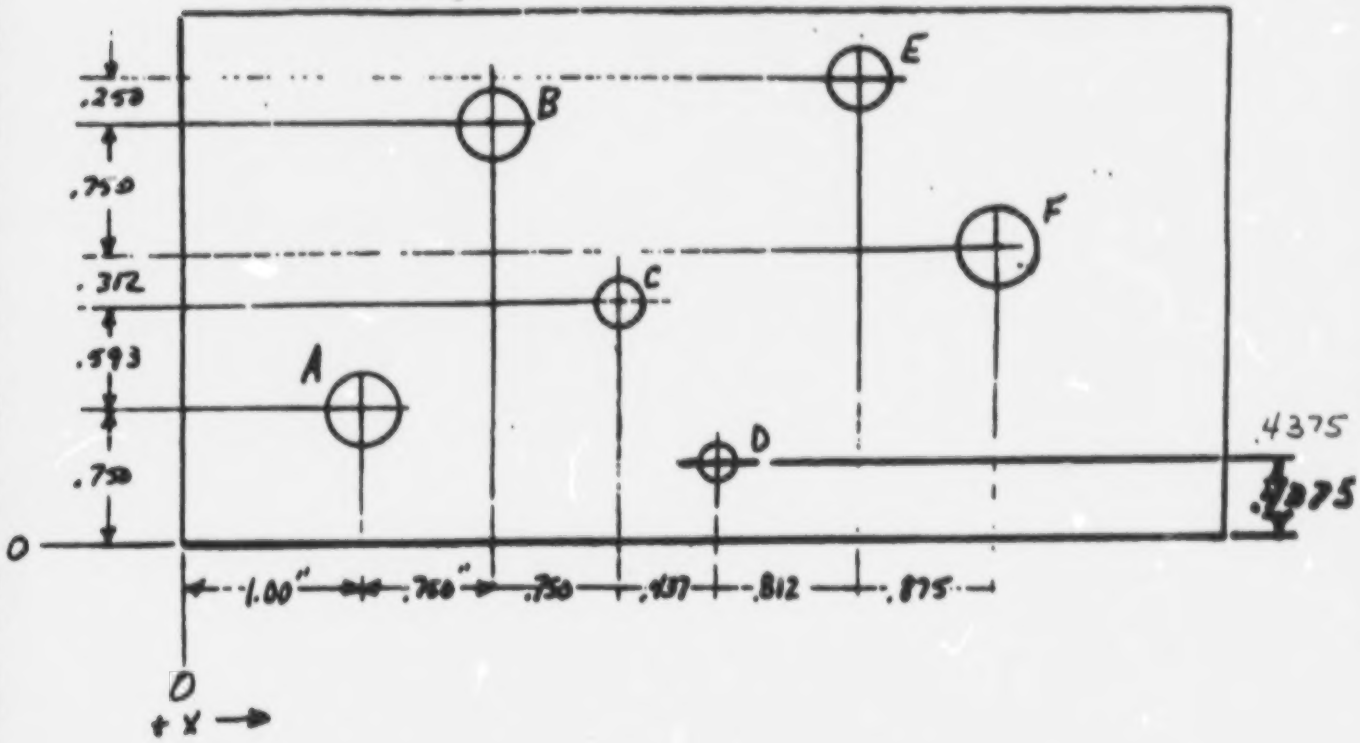

CNC DRAWING
"0" STOP
1.000
3.500
X
Y
-.500
-2.000
-5.000
#1
X1.000
Y-.500
#2
X3.500
Y-.500
#4
X1.000
Y-2.000
#3
X3.500
Y-2.000
#5
X1.000
Y-5.000
#6
X3.500
Y-2.000
ABSOLUTE
DIMENSIONS
"0" STOP
1.000
2.500
-.500
-1.500
-3.000
#1
X1.000
Y-.500
#2
X1.500
Y0.000
#4
X-1.500
Y0.000
#3
X0.000
Y-1.500
#5
X0.000
Y-3.000
#6
X2.500
Y0.000
INCREMENTAL
DIMENSIONS

MILWAUKEE AREA TECHNICAL COLLEGE

NC PROGRAMMING I

ASSIGNMENT UNIT 2

1. Give the X and Y values for points A-F absolutely and incrementally.



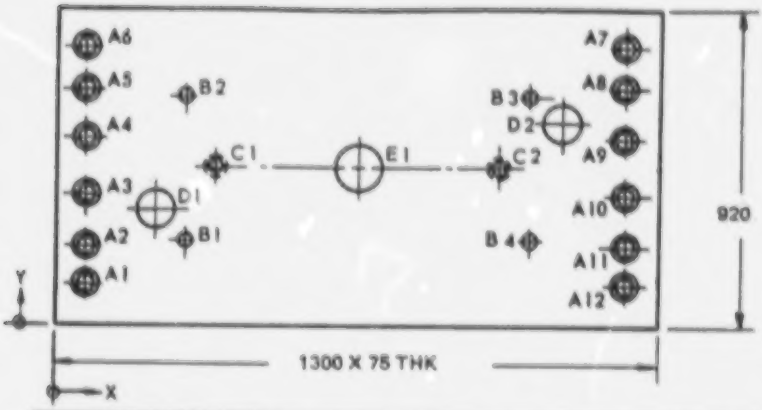
ABSOLUTE

A
B
C
D
E
F

INCREMENTAL

A
B
C
D
E
F

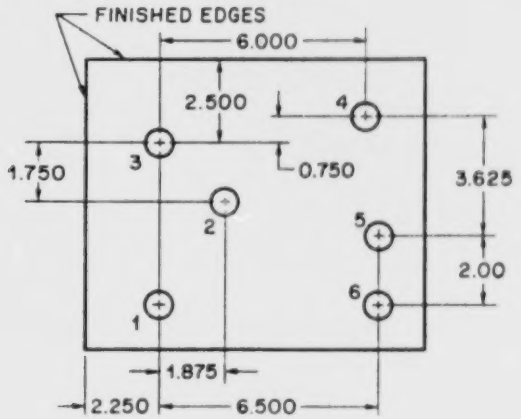
2. Plot the incremental position for all of the holes on this drawing, A1-A12, B1-B4, C1 and C2, D1 and D2, and E1 in that order. Use the locational information in the chart.



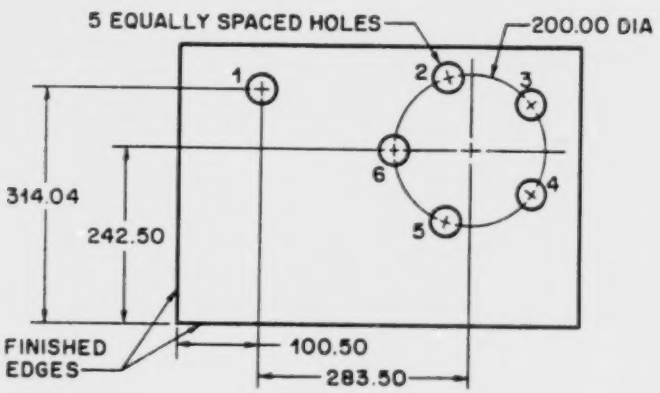
HOLE	LOCATION		HOLE SIZE
	X	Y	
A1	50	120	12X ⌀32
A2	50	235	⌴ ⌀64
A3	50	385	↧ 30
A4	50	535	NEAR SIDE ONLY
A5	50	685	
A6	50	800	
A7	1250	800	
A8	1250	685	
A9	1250	535	
A10	1250	385	
A11	1250	235	
A12	1250	120	

HOLE	LOCATION		HOLE SIZE
	X	Y	
B1	280	260	4X ⌀16
B2	280	660	
B3	1020	660	
B4	1020	260	
C1	342	460	2X M20
C2	958	460	
D1	210	334	2X $\phi^{80.030}_{80.000}$ (80 H7)
D2	1090	586	
E1	650	460	1X $\phi^{100.054}_{100.000}$ (100 H8)

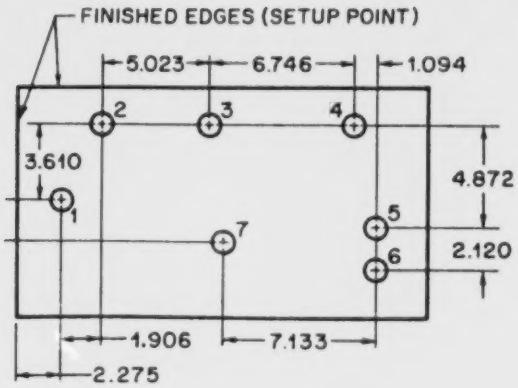
3. Plot the X and Y coordinates of holes 1-6 absolutely and incrementally in numerical order. Use the finished edges as part 0,0.



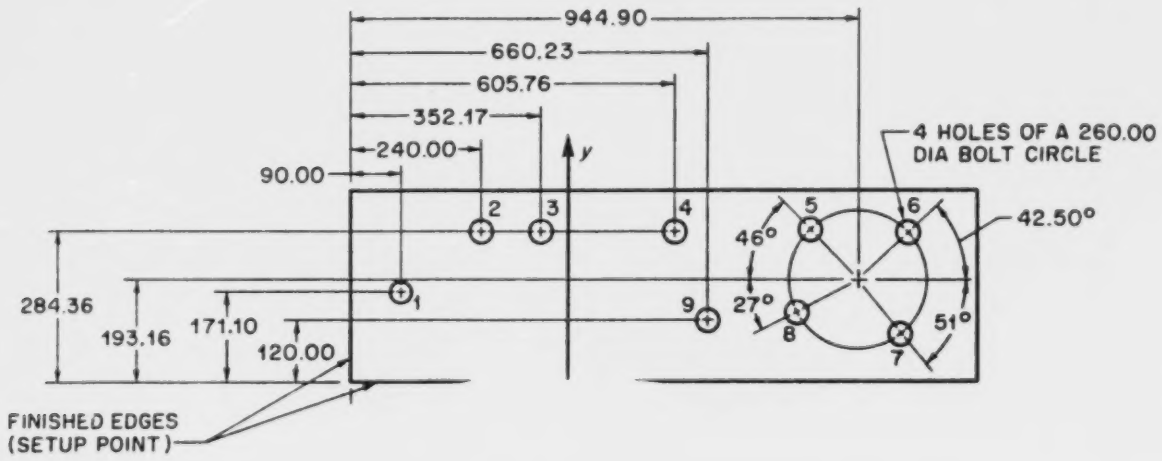
4. Plot the X and Y coordinates of holes 1-6 absolutely and incrementally in numerical order. You will have to use trigonometry to plot holes 2-6. If you do not know how to do this, see your instructor. Use the finished edges as part 0,0.



5. Plot the X and Y coordinates for holes 1-7 absolutely and incrementally in numerical order. Use the finished edges as part 0,0.



6. Plot the X and Y coordinates for holes 1-9 absolutely and incrementally in numerical order. Use the finished edges as part 0,0. Trigonometry is necessary for holes 5-8.



7. Plot the X and Y coordinates for holes A - M absolutely using the following values:

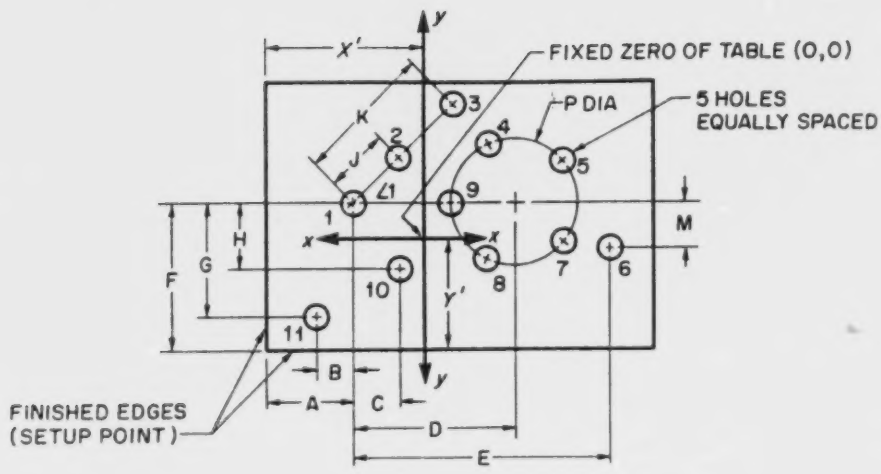
X'=-10 Y'=-8 A=5.175 B=1.300 C=3,250

D=14.25 E=22.00 F=12.50 G=9.15 H=5.15

J=5.50 K=11.25 M=4.625 PDia=10.00

Angle 1=42 degrees

Use the fixed table 0,0 indicated on the drawing as 0,0.



MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

UNIT 3 Basic Manual Programming

INTRODUCTION

In this unit we will begin to manually write CNC programs. In order to do this we must examine the G codes necessary for straight and circular tool movement.

INSTRUCTIONAL OBJECTIVES

When you have completed this unit you will be able to:

1. Plot coordinate points.
2. Correctly use G codes to write a basic program.

LEARNING MATERIALS

1. CNC Programming I, Information and Assignment Unit 3

LEARNING ACTIVITIES

1. Read CNC Programming I, Information Unit 3. (Objectives 1-2)
2. Complete CNC Programming I, Assignment Unit 3. (Objectives 1-2)

ASSIGNMENT

1. Read Information Unit 3.
2. Attend the Lecture/Discussions on Unit 3.
3. Complete Assignment Unit 3.

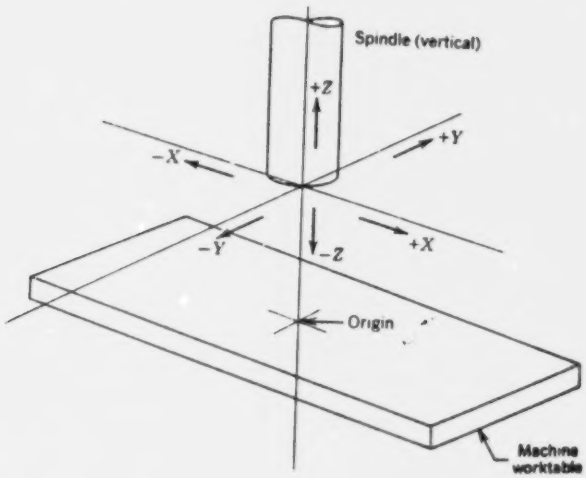
INFORMATION UNIT 3

So far we have seen how points can be identified on the Cartesian Coordinate grid and how points may be referred either to a common zero using absolute positioning (G90) or to a previous point using incremental positioning (G91). In this unit we will learn several more G codes and combine all of this information into a written CNC program.

The first G code we will describe is G0. This code is used to tell the machine to move at a very rapid rate, called rapid traverse. This rate, which can be anywhere from 200 IPM to 400 IPM depending upon the machine tool, is far too fast to be used to execute any machining moves. It is used to move the tool through the air rapidly in order to get to a machining or tool change position. A G code can be used to move the tool in either the X, Y or Z axis and it can be used to move either positively or negatively in all three axes.

The Z axis was referred to earlier but never described in depth. Z is the vertical axis. It is used to move the tool up away from the work (Z+) or down into the work (Z-). Just as with X and Y, the Z axis must have a 0 reference point. Z zero is usually set at the top of the work piece.

The following figure illustrates how all the axes relate to a CNC machine tool.



G0 should be used with caution because it move so rapidly. It should be used, however, whenever the tool is moving through the air. If an X, a Y, and a Z move are all on the same line, the machine will execute them simultaneously in a "ramping" type move. For this reason, we will put any G0 Z moves on separate lines.

The following are examples of how G0 is used in a program:

```
N5   G0   X0,Y0
```
(The "N5" is a line number which is used by programmers to identify the line. The machine does not read "N" data. In this line we are telling the machine to move at a rapid rate to the previously defined X and Y zero points.)

```
N10  G0   Z.100
```
(In this line we are telling the machine to move the tool straight down, at a rapid rate, to .100 above Z 0.)

The next G code is G1. G0 and G1 are the two most commonly used G codes. G1 is used to move the tool at a specified rate, called a feedrate, through the part. G1 is a machining move unlike G0 which is a rapid move. If X, Y, and Z moves are called for in a single line the machine will execute them simultaneously. Unless we wish to move at an angle or in a ramp, we never put X,Y,and/or Z on the same G1 line. In order for a G1 to occur, we must specify a feedrate. We do this by using a F code followed by the desired feedrate in IPM.

The following are examples of how G1 is used in a program:

```
N15  G1   Z-.100    F10
```
(This line tells the tool to move straight down .100 into the part at a rate of 10 IPM. The tool is now machining.)

```
N20  G1   X.500     F10
```
(This line tells the machine to machine .500 in the X+ direction - to the right of zero.)

The next two G codes are G2/G3. These are used to move the tool in a circular direction either clockwise (G2) or counterclockwise (G3). In order for these commands to be executed the following conditions must be met.

1. The tool should already have been moved to the beginning point of the circle.

2. The center of the circle must be defined using an I code for the X axis circle center location and a J code for the Y axis circle center location.

3. The X and Y locations given in the G2/G3 line should be the desired endpoints of the circular move.

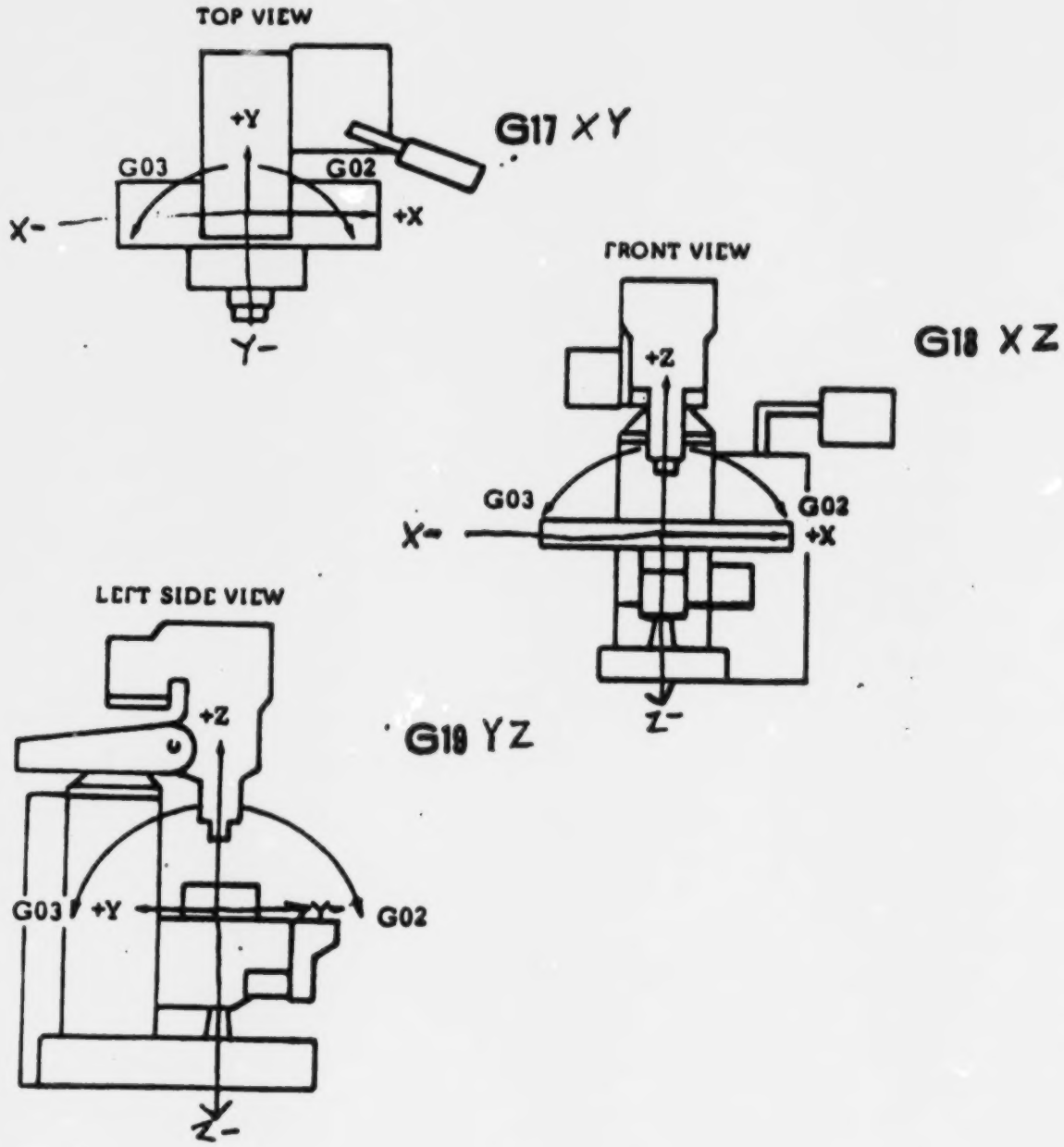
The following are examples of how a G2 or G3 could be used in a program:

```
N25  G2   X1.00,Y.500    I.250     J.250     F5
```

(This line will cause a clockwise circular move beginning with where the tool is already, ending at X1,Y.5, about a circle whose center is .250 to the right of X0 (I) and is .250 above Y0 (J) at a feedrate of 5 IPM. Since G2/G3 are machining moves, a feedrate must be specified.

Since circular moves can occur in a combination of any two axes, we must specify which two axes we wish. This is done using a G17, G18, or G19. G17 refers to the X and Y axes and will result in a flat circle. G18 refers to the X and Z axes and will result in raised or lowered circle parallel to the X axis. G19 refers to the Y and Z axes and will result in a raised or lowered circle parallel to the Y axis. These G codes are setup codes and will be entered at the beginning of a program. Only one of these three can be used at anyone time. G17 is the most common and we will use it.

The following figures illustrate how these codes relate to a machine tool.



The first program you will write in this class will be one which will tell the machine to spell out your initials or name. Here are the steps to follow in order to do this correctly. Refer to the "JOYCE" program and examples for each of these steps.

1. Carefully lay out the path of the tool (your name or initials) on graph paper. This tool path must have at least 2 circular moves. Assign some numerical value (.25 or ?) to each square on the graph paper. This will help with step 3.

2. Draw a box around your tool path to represent the edges of the piece we will machine. Assign a size to this piece of X5, Y4.

3. Accurately plot all of the coordinate points. These should include all start points, all direction change points, all circle/radius centers, and any end points. Label these points per the "JOYCE" example.

4. Use a T1 (tool number 1), a M25 (tool change), a S 800 (spindle speed of 800 RPM) and a F5 (feedrate of 5 IPM) per the "JOYCE" example.

5. Write the entire program down line by line beginning with N5 and moving up by increments of 5.

6. Proofread the program line by line.

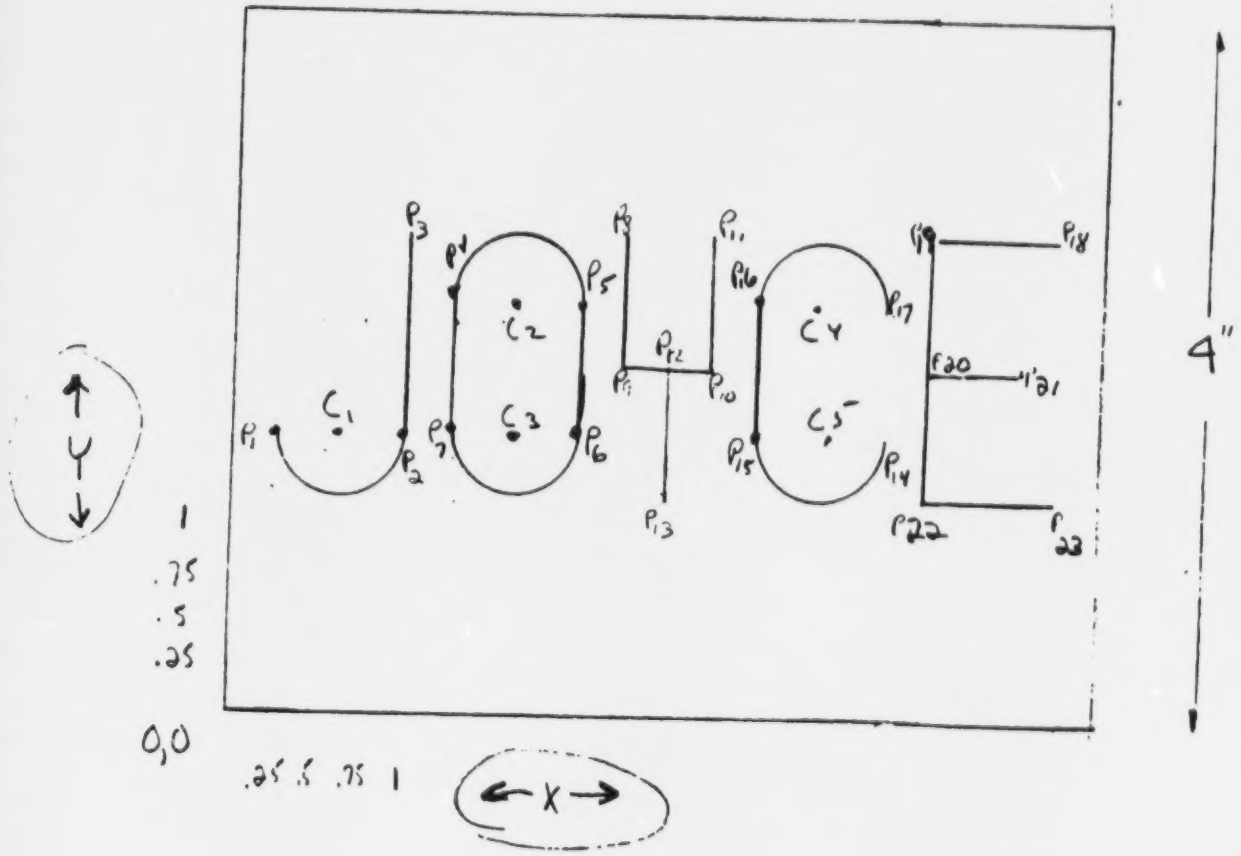
Use the "JOYCE" example which is found on the next two pages.

	X, Y
P_1	.25, 1.625
P_2	1, 1.625
P_3	1, 2.75
P_4	1.25, 2.375
P_5	2.00, 2.375
P_6	2.00, 1.625
P_7	1.25, 1.625
P_8	2.25, 2.75

	X, Y
P_9	2.25, 2.00
P_{10}	2.75, 2.00
P_{11}	2.75, 2.75
P_{12}	2.50, 2.00
P_{13}	2.50, 1.25
P_{14}	3.75, 1.625
P_{15}	3.00, 1.625
P_{16}	3.00, 2.375

	X, Y
P_{17}	3.75, 2.375
P_{18}	4.75, 2.75
P_{19}	4.00, 2.75
P_{20}	4.00, 2.00
P_{21}	4.50, 2.00
P_{22}	4.00, 1.25
P_{23}	4.75, 1.25

C_1	.625, 1.625	C_4 3.375, 2.
C_2	1.625, 2.375	C_5 3.375, 1.
C_3	1.625, 1.625	



```
        N5    G90, G70
        N10   G40 ,G0  X0, Y0, (AUX) M25
        N15   (AUX)  S 800 T1
P1      N20   G0  X .25", Y 1.625, Z.1  (P1)
        N25   G1    Z-.025" F 5 (X abs)  (Y abs)
P2      N 30  G3 G45 X1, Y 1.625 (P2) I .625  J 1.625 (C1)
P3      N 35  G1  X1, Y2.75  F5 (P3)
P4      N 40  G0  X 1.25" Y 2.375", Z.1 (P4)
        N 45  G1  Z-.025  F5
P5      N 50  G2  X 2.00, Y 2.375, I 1.625"  J 2.375" (C2)
P6      N 55  G1  X 2.00  Y 1.625"  F5
P7      N 60  G2  X 1.25, Y 1.625" I 1.625"  J 1.625" (C3)
P4      N 65  G1  X 1.25", Y 2.375"  F5
P8      N 70  G0  X 2.25", Y 2.75", Z .1
        N 75  G1  Z-.025   F5
P9      N 80  G1  X 2.25"  Y 2.0
P12     N 85      X 2.50   Y 2.0
P13     N 90      X 2.50   Y 1.25"
P.a     N 95      X 2.50   Y 2.0
P10     N 100     X 2.75"  Y 2.0
P.      N 105     X 2.75"  Y 2.75"
P11     N 110  G0  X 3.75   Y 2.375"  Z .1
        N 115  G1  Z-.025   F5
P16     N 120  G3  X 3.00   Y 2.375"  I 3.375  J 2.375" (C4)
P15     N 125  G1  X 3.00   Y 1.625  F5
P14     N 130  G3  X 3.75   Y 1.625"  I 3.375  J 1.625" (C5)
P18     N 135" G0  X 4.75   Y 2.75  Z.1
        N 140  G1  Z-.025   F5
P19     N 145  G1  X 4.00   Y 2.75
P20     N 150  G1  X 4.00   Y 2.00
P21     N 155  G1  X 4.50   Y 2.00
P.0     N 160  G1  X 4.00   Y 2.00
P22     N 165  G1  X 4.00   Y 1.25"
P23     N 170  G1  X 4.75"  Y 1.25"
        N 175  G0  Z.1                    32
        N 180  M22
```


BEST COPY AVAILABLE

So far we have looked at some of the most common G codes. This list is by no means complete. There are hundreds of G codes. Most machines recognize these as standard although there are some discrepancies between CNC controllers as to G code recognition. All of the ones we will use are standard and would be recognized correctly by any CNC controller.

The following is a list of G codes as recognized by a common CNC controller.

G CODE	GROUP	FUNCTION
0	1	Rapid traverse
1	1	Linear interpolation (feed)
2	1	Circular interpolation CW
3	1	Circular interpolation CCW
4	0	Dwell
8	11	Modal deceleration override off
9	11	Modal deceleration override on
12	0	Helical interpolation CW
13	0	Helical interpolation CCW
17	2	XY plane selection
18	2	ZX plane selection
19	2	YZ plane selection
22	0	Circular interpolation, fillet input CW
23	0	Circular interpolation, fillet input CCW
30	3	Mirror Image off
31	3	Mirror Image X on
32	3	Mirror Image Y on
40	4	Cutter diameter offset off
41	4	Cutter compensation left
42	4	Cutter compensation right
44	5	Cutter compensation, normal feed-rate
45	5	Cutter compensation, modify feed-rate
70	6	Input in inch
71	6	Input in millimeter
72	7	Transformation off
73	7	Transformation/Rotation, Scaling
74	8	Multi quadrant circle input off
75	8	Multi quadrant circle input on
77	1	Zig Zag Mill cycle
78	1	Pocket Mill cycle
79	1	Bore Mill cycle
80	1	Drill cycle off
81	1	Z cycle, drill (feed in, rapid out)
82	1	Z cycle, spot face (feed in, rapid out)
83	1	Z cycle, deep hole (peck, rapid out)
84	1	Z cycle, tap (feed in, feed out)
85	1	Z cycle, bore (feed in, feed out)
86	1	Z cycle, bore (feed in, stop-wait, rapid out)
87	1	Z cycle, chip break (peck, rapid out)
89	1	Z cycle, bore (feed in, dwell, feed out)

90	9	Absolute programming
91	9	Incremental programming
92	0	Preset part programming zero point
96	10	Restore base part program coordinate system
97	10	Set work coordinate system
99	0	Deceleration override
170	1	Outside Frame Mill
171	1	Inside Frame Mill
172	1	Pocket Frame Mill
173	1	Outside Face Mill
174	1	Inside Face Mill
175	1	Outside Circle Mill
176	1	Inside Circle Mill
177	1	Pocket Circle Mill
179	1	Slot Mill
181-189	1	Z cycle (same as G81-G89) multi-hole
191-199	1	Z cycle (same as G81-G89) frame of holes

The initial start up or clear settings of the preparatory functions are:

G CODE	GROUP	FUNCTION
0	1	Rapid traverse 250 IPM
8	11	Modal deceleration override off
17	2	XY plane selection
30	3	Mirror Image off
40	4	Cutter compensation off
45	5	Cutter compensation, normal feed-rate
70	6	Input in inch
72	7	Transformation off
75	8	Multi-quadrant circle input on
90	9	Absolute programming
96	10	Base coordinate system

Only 1 G code from each of 12 groups active at any one time.

MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

ASSIGNMENT UNIT 3

1. Write a program, either one of your own (spouses' name, children's name, boss's name, etc.), or one assigned by the instructor. This program should be linear and circular moves. Consult with the instructor regarding feeds, speeds, tools to be used, and Z depths.

MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

UNIT 4 Polar Coordinates and Cutter Compensation

INTRODUCTION

In this unit two additional concepts will be introduced. These are the use of polar coordinates to define points and the use of cutter compensation within a CNC program.

INSTRUCTIONAL OBJECTIVES

When you have completed this unit you will be able to:

1. Correctly define a point using polar coordinates.
2. Incorporate polar positioning into a CNC program.
3. Calculate cutter compensation manually.
4. Correctly incorporate cutter compensation into a CNC program manually.
5. Correctly incorporate cutter compensation into a CNC program using G40, G41, and G42.

LEARNING MATERIALS

1. CNC Programming I, Information and Assignment Unit 4.

LEARNING ACTIVITIES

1. Read CNC Programming I, Information Unit 4. (Objectives 1-5)
2. Complete CNC Programming I, Assignment Unit 4. (Objectives 1-5)

ASSIGNMENT

1. Read Information Unit 4.
2. Attend the Lecture/Discussion on Unit 4.
3. Complete Assignment Unit 4.

INFORMATION UNIT 4

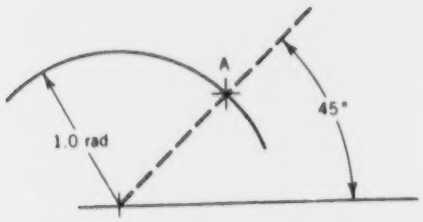
There are two different ways to define the location of a point and so far in this book we have looked at only one of them. In this unit we will learn how to use the other.

The two ways of defining a point are by rectangular coordinates, that is the intersection of two lines and by polar coordinates, the intersection of a radius at a certain angle with the circumference of a circle. Rectangular coordinate positioning is the method we have used thus far. It defines a point as the intersection of two lines, one a certain distance from the X axis and the other a certain distance from the Y axis. This intersection point is given as a numerical value representing those distances. (X4, Y3) means that the intersection point is 4 units in the X positive direction and 3 units in the Y positive direction. This type of positioning works very well with straight line dimensions but doesn't work so well for points on a circle. You may recall that in Unit 2 we had to use trigonometry to calculate the location of some points on a circle.

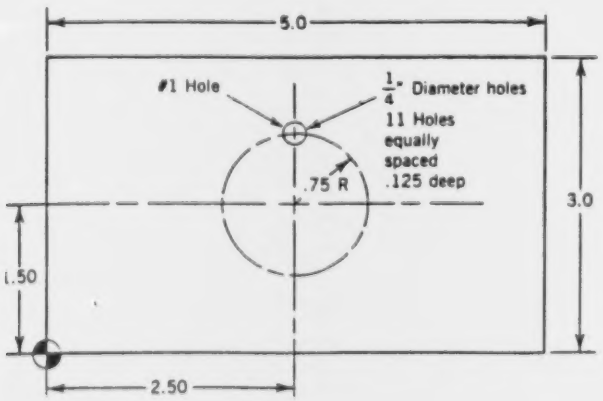
Polar positioning works very well for defining points found on a circle. This is because it uses circles to define the location of points. Polar coordinates are the intersection of a radius swung at a specified angle with the circumference.

Polar coordinates are put of and refer to the Cartesian coordinate system just as are rectangular coordinates. For example, the center of the circle is defined as a rectangular coordinate in relation to a 0,0 point. Also the angle of the radius is given as a number of degrees, either positive (counterclockwise) from zero degrees (3 o'clock) or negative (clockwise) from zero degrees (3 o'clock). These parameters are all part of the Cartesian system.

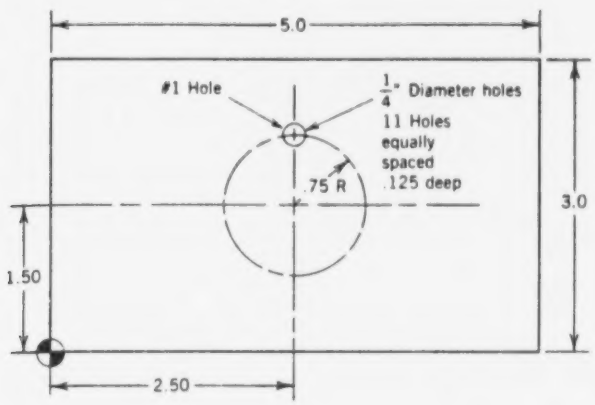
The following figure illustrates how the polar coordinate system is used. Instead of using trigonometry to calculate the X and Y coordinates of point A, we can define it as a point swung at 45 degrees by a one inch radius on the circumference of a circle. Since a point is defined geometrically as the intersection of two lines, we have defined a point. There is only one place that a one inch radius swung at 45 degrees will intersect that circle, Point A.



The only thing we needed to know about the above example that we didn't have was the center point of the circle. It does no good to define a point on an angle of a circle if we don't know where that circle is. The following example illustrates how polar coordinates are used in a CNC program. We would define the #1 hole as a point at 90 degrees swung by a .75 radius on a circle whose center is at X 2.50, Y 1.50.



How would we go about defining the rest of the 11 equally spaced holes? That phrase, equally spaced is the key. Since we have 11 holes, equally spaced, we know that they are the same number of degrees apart. If we divide 360 (the number of degrees in a circle) by 11 we will know how far apart the holes are. That spacing is 32 degrees, 43 minutes, and 38 seconds. If hole #1 is at 90 degrees, then hole #2 moving counterclockwise is at 90 degrees + 32 degrees, 43 minutes, 38 seconds. Thus hole # 2 would be at 122 degrees, 43 minutes, 38 seconds. This example is correct if we are programming absolutely. If we are programming incrementally, each hole will be 32 degrees, 43 minutes, and 38 seconds from the previous hole (0 degrees).



Using the above example, find the polar coordinates for holes 3-11 absolutely.

Each CNC controller will allow you use polar coordinates somewhat differently. The steps are usually these:

1. You must tell the machine that you are going to be using polar coordinates. This is usually done by pushing a "POLAR" key or button.

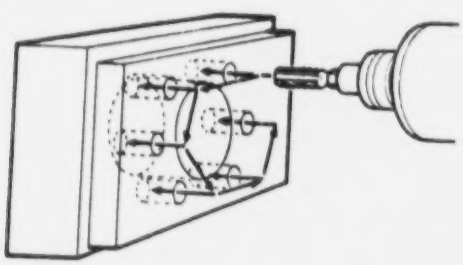
2. You must define the circle by giving it center location as an I (X) coordinate and as a J (Y) coordinate.

3. You must define the size of the circle as a radius or a diameter.

4. You must specify the angular rotation about that circle as a number of degrees.

The second concept we are going to discuss in this unit is cutter compensation. This is used to shift the location of the spindle a specified amount in order to allow the cutting edge of the machine's tool to engage the work at the correct dimension.

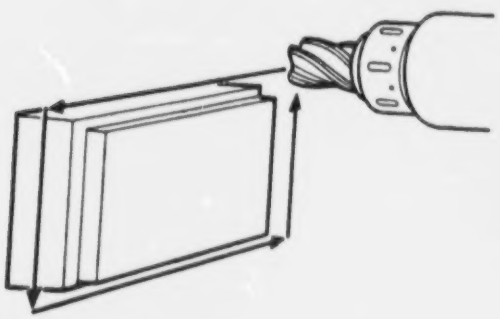
Unless told otherwise, the machine tool will interpret all dimensions as spindle centerline dimensions. Sometimes this is the correct interpretation. When we programmed the machine to use a felt tip pen to spell out letters, we wanted the center of the machine's spindle (the center of the pen) to be at the dimensions we entered. Also when drilling or tapping holes, the center of the spindle should be on the center of the hole locations. The following figure illustrates this.



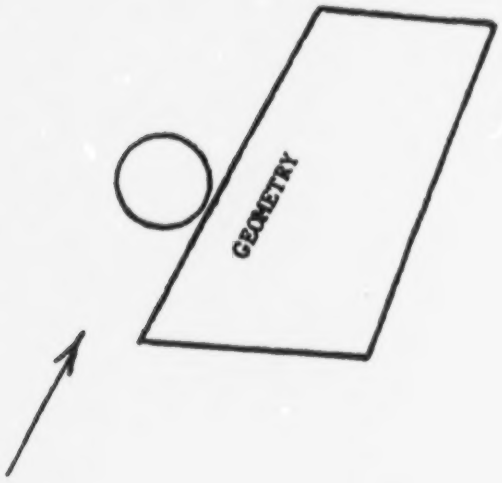
Many times this is not what we want. When using the perimeter of a tool to machine sidewalls, either outside or inside, we want the edge of the cutter to be at a certain point so we must shift the spindle centerline away from the point to compensate for the radius size of the cutter. In order to cause this shift to occur, either left or right, we must do one of two things.

The first method to compensate for cutter radius is to recalculate the coordinate values, either adding or subtracting the cutter radius. When the machine moves to the adjusted positions the edge of the cutter will be in the desired position due to the cutter radius adjustments.

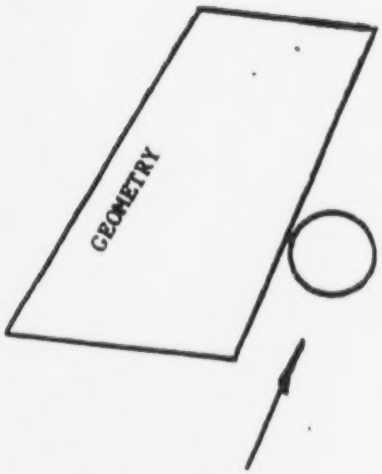
In order to allow the edge of the endmill in the figure below to engage the work at the correct position we must shift the center of the spindle by 1/2 the cutter's diameter (cutter radius). In order to shift the center of the spindle to the right we must either add or subtract the cutter radius from the workpiece X and Y values.



It can be confusing trying to decided whether to shift the cutter to left or to the right. The best way to understand this is to visualize yourself "riding" on the cutter facing the direction that the cutter is being fed through the work. As you ride along on this cutter, which way should you be shifted from the work? If you and the cutter are would be shifted to the left, then use that direction in your calculations. If you are shifted to your right then use that direction. Shift direction is tied directly to feed direction. You cannot calculate cutter compensation without first knowing feed direction! The following figures illustrate how cutter compensation direction is tied to feed direction.



Cutter Compensation Left

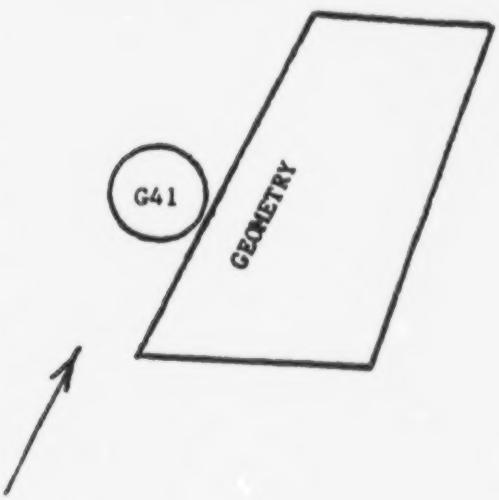


Cutter Compensation Right

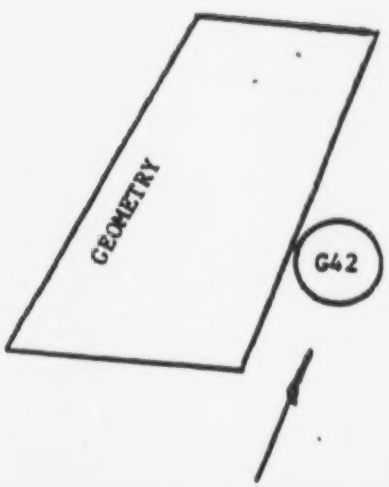
The second, and more efficient, way to incorporate cutter compensation into a CNC program is to use the cutter compensation G codes. These are G40 - cutter compensation off, G41 - cutter compensation left, and G42 - cutter compensation right. When a G41 or a G42 are in effect, the controller will automatically shift the spindle centerline the radius value of the tool it is using in the appropriate direction.

How does the controller know what tool it is using and how much to shift? It does this by referring to a "Tool Library". This is a record, stored by the programmer in the machine's memory, of certain information about the tools being used. The tools will be labeled by "T" numbers; T1 is tool number 1, etc. Included in this information is the tool's diameter. Once a G41 has been entered into the program, the controller will shift all tools 1/2 their stored diameter to the left. It is very important that the correct data be stored in the tool library and that the correct tool is being used!! Mistakes in this area can be very costly.

The following figure shows how a G41 and G42 can influence cutter compensation.



Cutter Compensation Left



Cutter Compensation Right

. Because we are relying on the controller to cause cutter compensation to occur, it is very critical that we know when it is in effect and when it isn't! G40 will turn any other cutter compensations off. It is a good idea to include a G40 in the beginning of a program. G41 will turn all other cutter . compensations off and will remain in effect until it is cancelled with either a G40 or G42. The same is true for G42. These three cutter compensations will cancel each other out and one must be in effect for a program to be executed.

MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

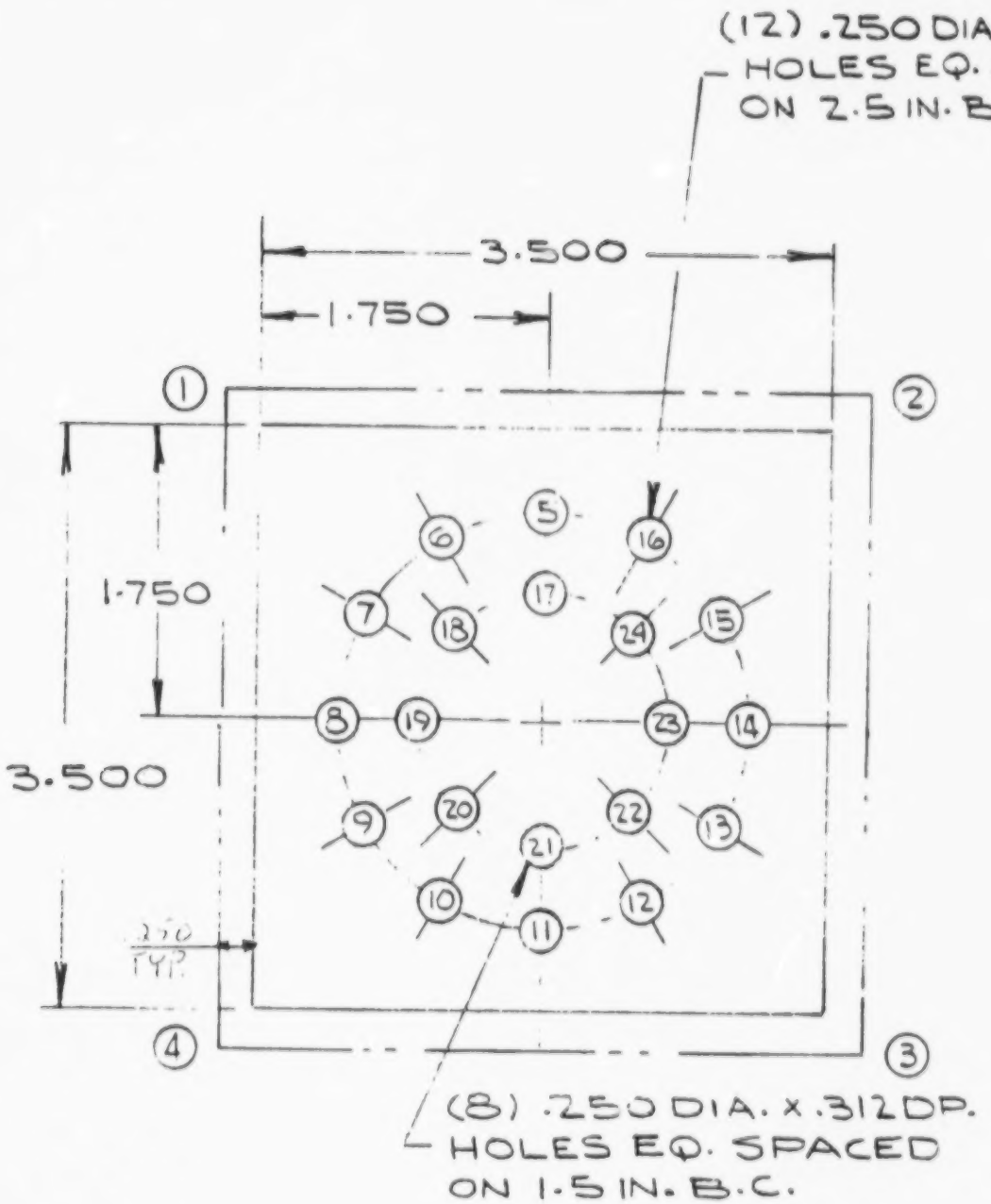
ASSIGNMENT UNIT 4

1. Calculate the absolute polar coordinates for holes 2-6 from Question #4 in Unit 2.

2. Calculate the absolute polar coordinates for holes 5-8 from Question #6 in Unit 2.

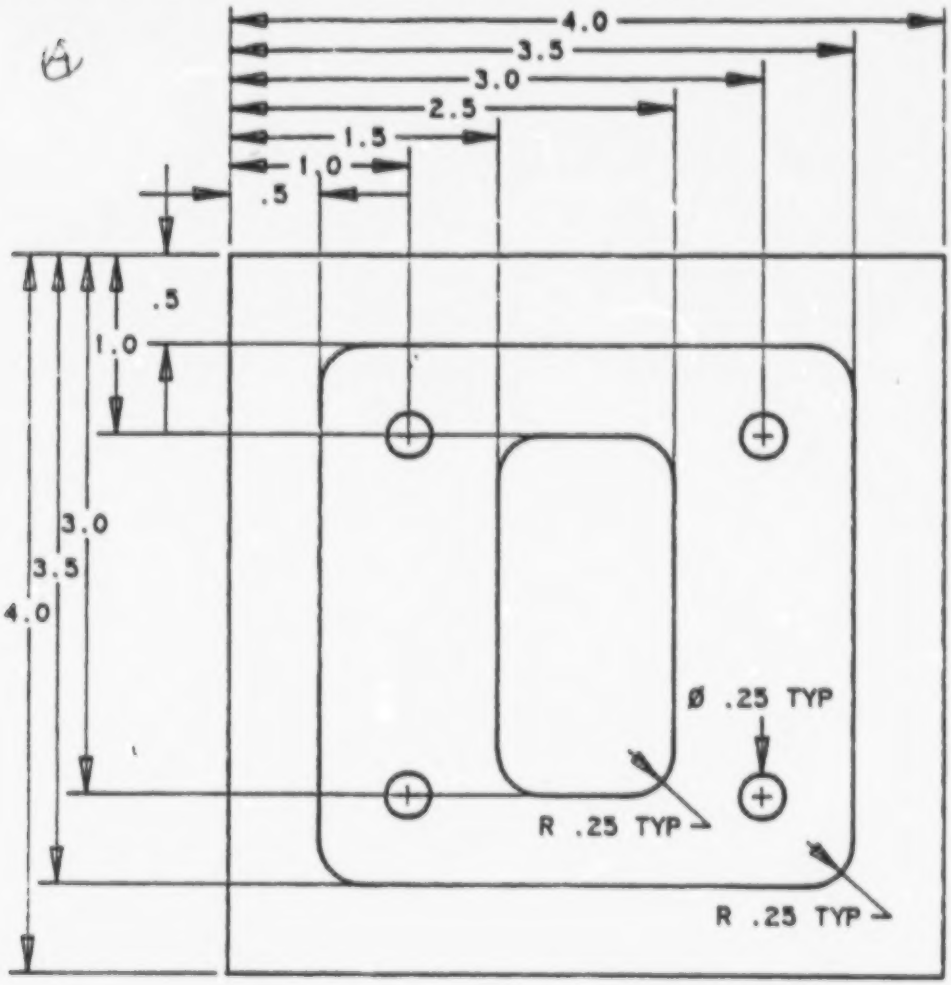
3. Calculate the absolute polar coordinated for holes 4-9 from Question #7 in Unit 2.

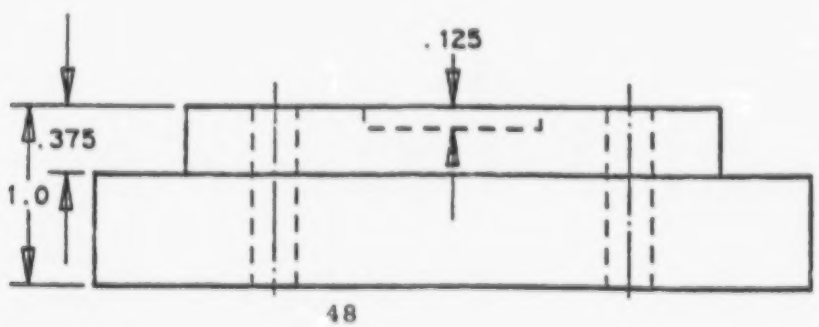
4. Plot the coordinates of the holes 5 - 24 using absolute polar coordinates. Plot them in numerical order using incremental polar coordinates. Use the upper left corner of the piece as 0,0.



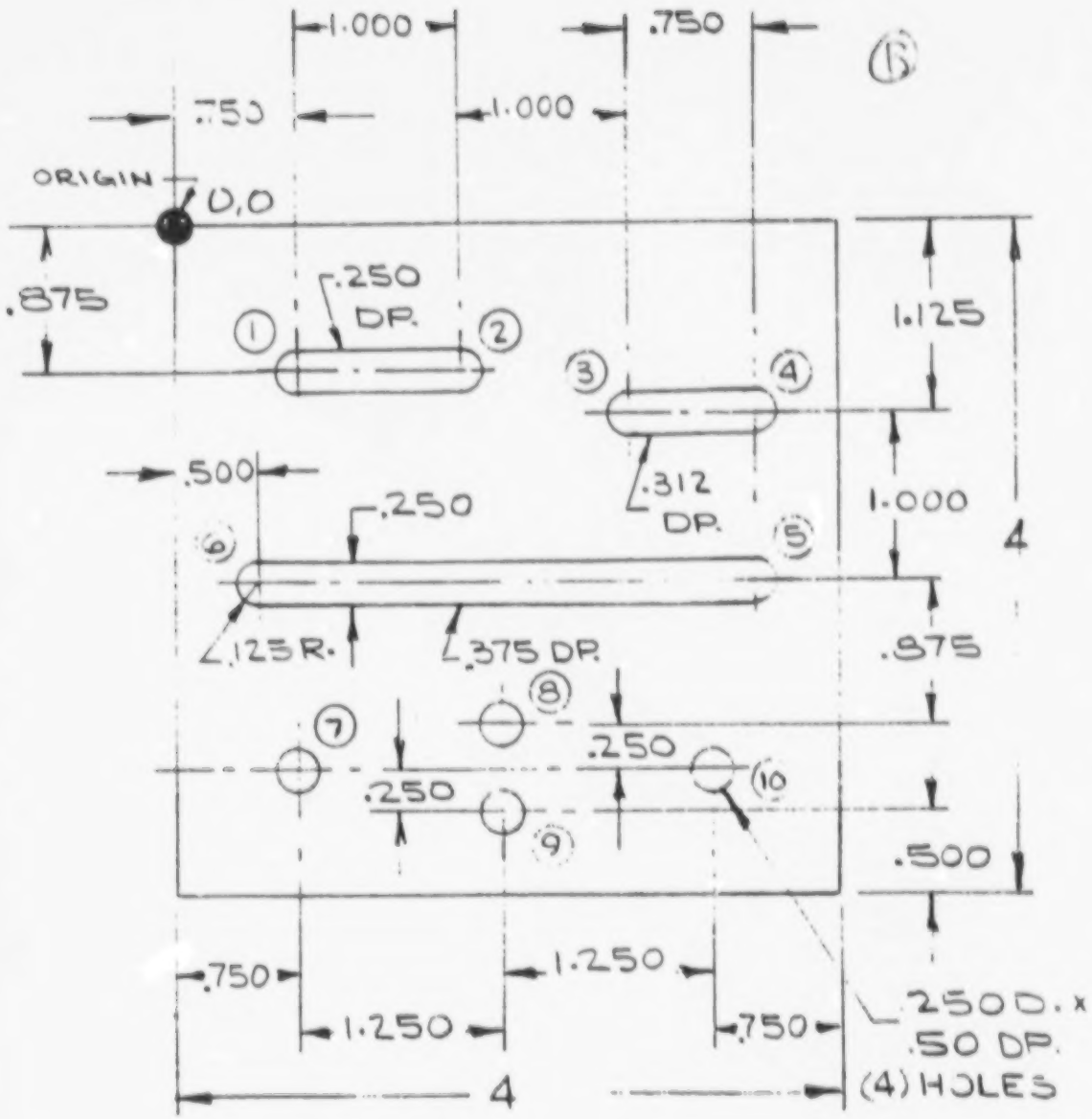
BEST COPY AVAILABLE

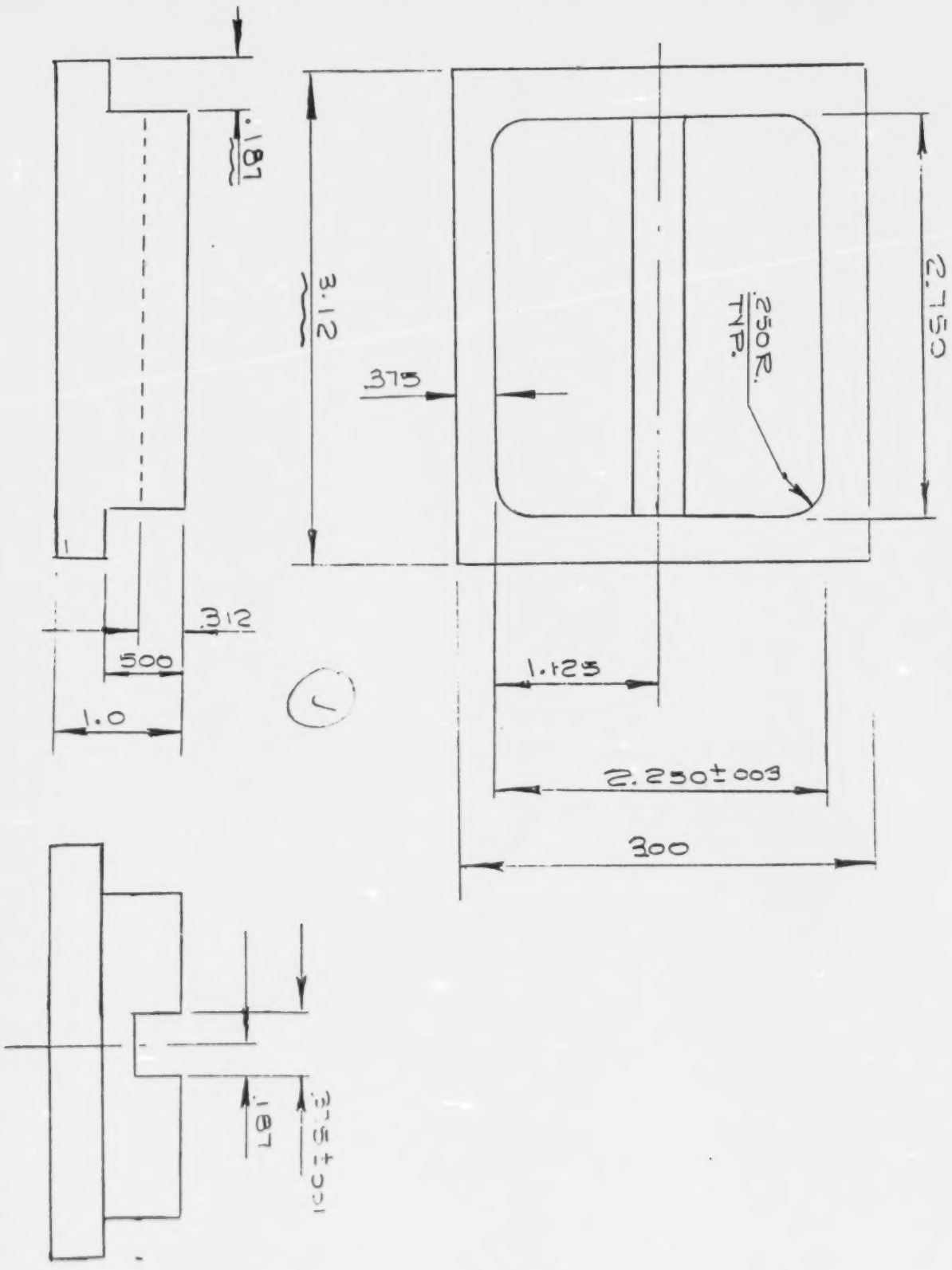
5. Write a CNC program for each of the three piece parts which follow. Consult with the instructor regarding 0,0 point (unless specified on the drawing), tool diameters, speeds and feeds. Write the programs twice, first using G40 (manually calculating cutter compensation) and then using either G41 or G42 as they would be needed.





48


1.000
.750
.750
1.000
ORIGIN
0,0
.875
.250
DP.
1
2
3
4
1.125
.500
.250
.312
DP.
1.000
6
5
4
.125 R.
.375 DP.
.875
8
7
.250
.250
10
9
.500
.750
1.250
1.250
.750
.250 D. x
.50 DP.
(4) HOLES
4


.187
3.12
.375
.250R.
TYP.
2.750
.312
.500
1.0
1.125
2.250±.003
300
.187
.375±.001

MILWAUKEE AREA TECHNICAL COLLEGE

CNC PROGRAMMING I

UNIT 5 Programming the Heidenhain Controller

INTRODUCTION

In this unit we will learn how to program a specific CNC controller, the Heidenhain. This controller uses a "Dialog" type of format which is different from the format we have been using. All controllers are different and this one has its own unique characteristics.

INSTRUCTIONAL OBJECTIVES

When you have completed this unit you will be able to:

1. Start up a milling machine equipped with a Heidenhain controller.
2. Set zero reference points using a Heidenhain controller.
3. Define a block on the Heidenhain controller.
4. Enter a program on the Heidenhain controller.
5. Verify a program using the Heidenhain graphics capabilities.

LEARNING MATERIALS

1. CNC Programming I, Information and Assignment Unit 5.
2. Bridgeport Programming and Operating Manual.

LEARNING ACTIVITIES

1. Read CNC Programming I, Information Unit 5. (Objectives 1-5)

2. Read Bridgeport Programming and Operating Manual, Chapters 1 and 2. (Objectives 1-5)

3. Complete CNC Programming I, Assignment Unit 5. (Objectives 1-5)

ASSIGNMENT

1. Read Information Unit 5.

2. Read Chapters 1 and 2 in the Bridgeport Programming and Operating Manual.

3. Attend the Lecture/Discussions on Unit 5.

4. Complete Assignment Unit 5.

INFORMATION UNIT 5

The Heidenhain Control is a 3 axis (X, Y, and Z) manual data input (MDI) control. It was designed so that the operator can enter a program directly at the machine. This is done by using what is called a "Dialog" format. That is, the control asks questions via prompts on the screen and the operator responds with the appropriate data. Thus it is very important that all of the coordinate points be plotted, tool definitions be decided upon, and machining sequences be determined before entering a program at the machine.

1. START UP

   a. Turn the power on at the rear of the machine.

   b. Engage the power enable button.

   c. Use the cycle start button to "move the machine over all 3 axis". There will be a prompt on the screen telling you to do this. This must be done twice in order to allow the control to reference itself to its "home" positions.

2. SETTING ZEROS

   a. Use the button shaped like a handwheel to call up the manual operations mode. In this handwheel mode you may use either the joysticks or the electronic handwheel for machine movement.

   b. Using the joysticks - the X and Y axes are controlled by the one to the lower right of the table and the Z axis in controlled by the one on the head of the machine - will cause the machine to move in rapid traverse in the selected axis and direction.

   c. Using the handwheel, also to the lower right of the table, will move the machine at a preselected feedrate. The feedrate is established by keying in an interpolation factor. The screen will be prompting for this factor. The higher the number, the slower the feedrate. A factor of 2 =.197"/handwheel rev., a factor of 3= .118/rev, a factor of 4=.05/rev, a factor of 5=.025/rev, factor 6=.020/rev, factor 7=.012/rev, 8=.003/rev, 9=.0015/rev, and 10=.0008/rev.

d. Use the necessary combination of joystick and handwheel to indicate your part straight on the table and to establish X and Y zero. The axis you wish to move in is selected by highlighting the appropriate axis on the screen with the cursor. When you have reached your zero position, enter a 0 into the appropriate axis on the screen by pushing the 0 key and then enter.

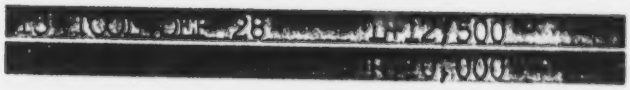
e. Set Z zero (Tool Length Offsets) using the following method. Put the longest tool in the spindle first. Call up TOOL DEFINITION and give the tool a number according to the order it will be used in the program (#1 will be used first, etc.). Bring the longest tool down carefully until it just touches the work. Enter a length of 0 for that tool (L0) and set Z zero at that point. Touch each of the other tools on the work and set their length at whatever Z position is on the screen, being sure to enter either a positive or negative Z value as is required.

f. The tool size should also be entered in TOOL DEFINITION. Note that we enter the tool radius as a "R" value.

The following 2 pages illustrate how tool lengths and radii are defined as well as how a tool is called up into a program. Notice that all of the dimensions are in Millimeters. If you wish to convert MM to decimal inches, simply divide MM by 25.4 and the answer will be in decimal inches.

TOOL DEF

Tool definition

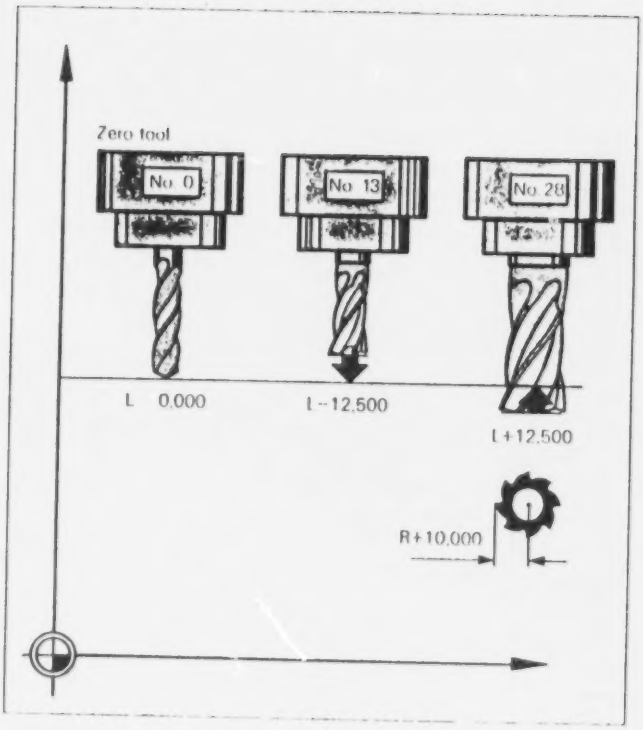


Tool length compensation:

Compensation values are referenced to the "zero tool".

Tools longer than the zero tool:
+ sign

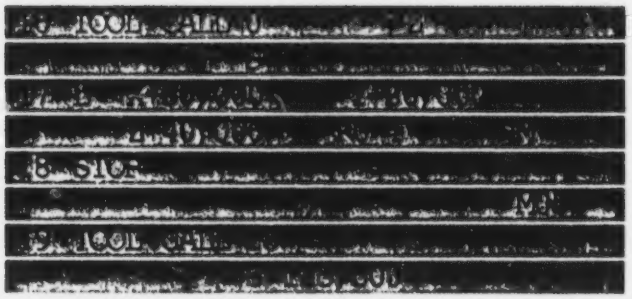
Tools shorter than the zero tool
– sign



BEST COPY AVAILABLE

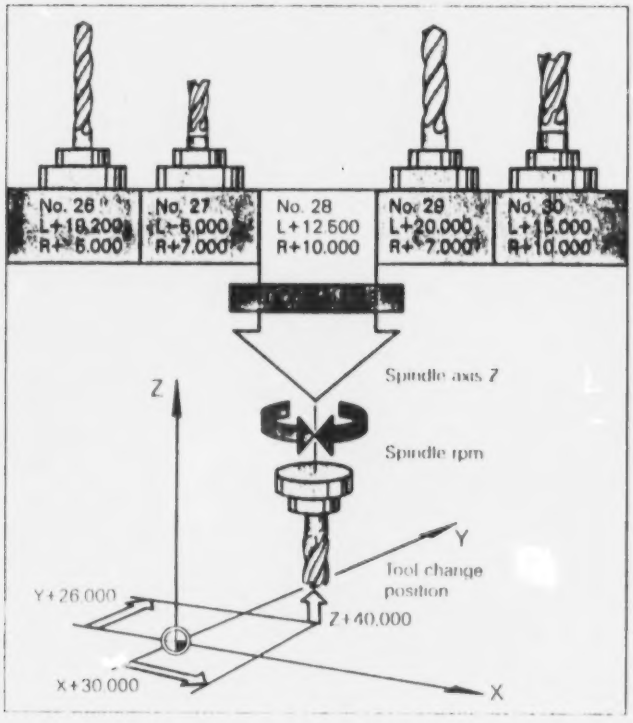
Tool call

TOOL CALL



Cancellation of current tool compensation values
Tool change position

Program run STOP (for manual tool change)
Call up of next tool with appropriate compensation values

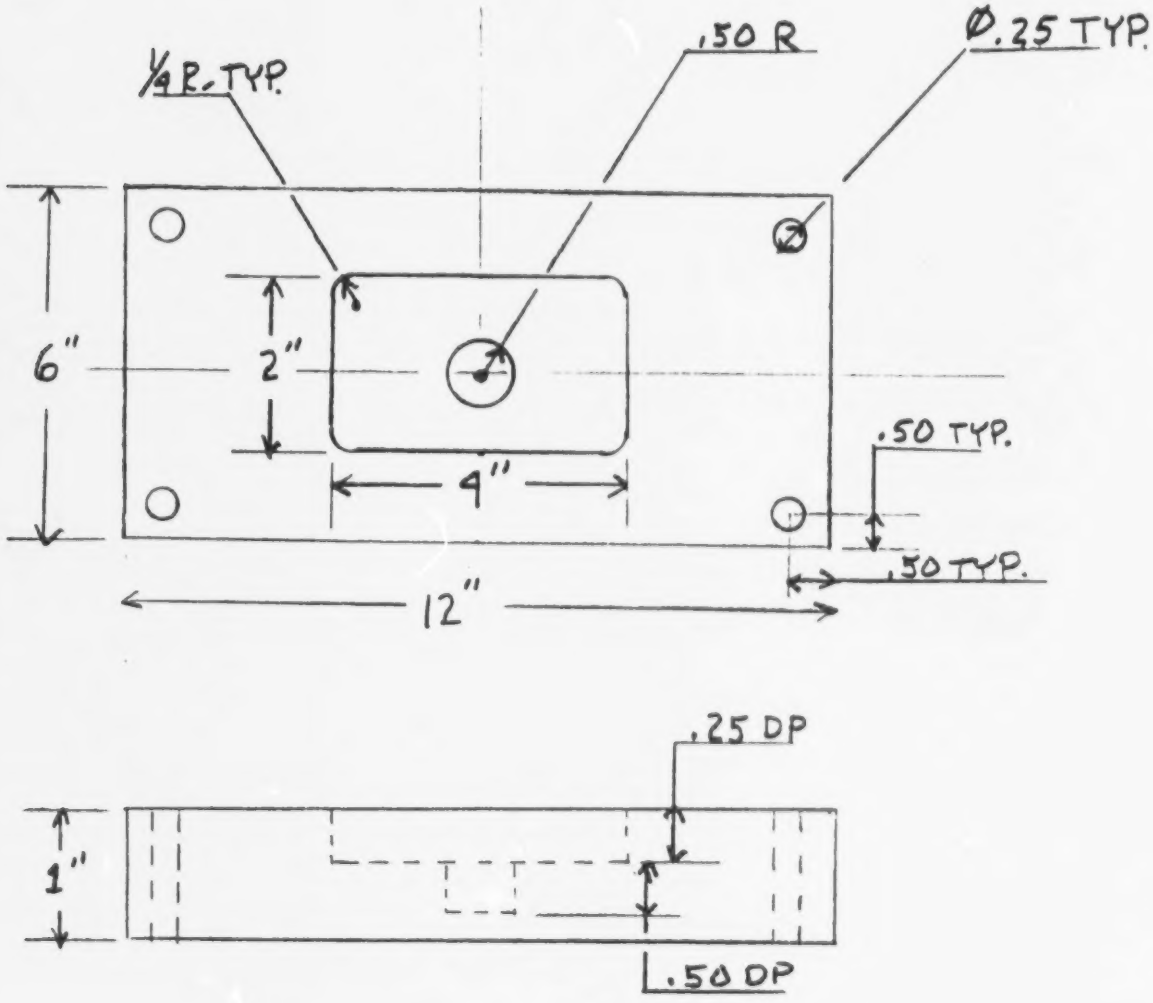


When traversing to the tool change position, take care that the tool does not collide with the workpiece!

The rotating direction of the spindle is determined by an auxiliary function: M03/M04 or M13/M14 (right/left)

BEST COPY AVAILABLE

Now lets take a look at a part print and how the program which will result in that part is entered into the Heidenhain control. Please refer to the following drawing:



This program assumes that X and Y zero have been set and that the proper tool lengths have been entered. After all of this has been done:

1. Call up the EDIT mode.
2. Assign a program number.
3. Give the proper tool numbers, lengths and radii.

This is the program for the preceding part.

```
0    Begin Program        Program Number "1? 7"
1    Tool Def 1           L ?  R.5
2    Tool Def 2           L ?  R .25
3    Tool Def 3           L ?  R .125
```

(We have now established the lengths and the radii of the three tool for this job, a 1" and a 1/4" diameter end mill and a 1/2" drill. See page 54.)

```
4    Blk Form 0.1         X-1  Y-1  Z-1
5    Blk Form 0.2         X13  Y7   Z0
```

(These two lines establish the size of the rough piece so that the machine will be able to demonstrate the machining cycle in the graphics mode. I made the rough block 1"/side larger than the finished piece so that the graphics would show the outside machining better. We define a block by giving the X,Y,and Z points of the lower left and upper right corners.)

```
6    Stop           M25  (This is a tool change.)
7    Tool Call 1         S4000      (Tool #1 , Spindle speed of
                                    4000 RPM.  See page
                                    55.)

8    L (linear)     X-1, Y-1,       R0   F2000
```

(This calls for a rapid linear move to the specified coordinates. There is no real G0 type rapid traverse. In order to rapid a feedrate of 200.0 must be given. R0 turns cutter comp off. Notice the assumed decimal point 1 place to the right in feedrates only! See page 60.)

```
9    L          Z.1        (Rapid to .100 above part. Feedrate is
                            modal as long as either "NO ENT" or "END
                            BLOCK" are used to end a line.)
10   L          Z-.1       F100
11   L    X0,Y0      RL    F100       (RL turns on cutter comp
                                      left.)
12   L    Y6
13   L    X12
14   L    Y0
```

```
15   L    X0
16   L    Y-1
17   STOP            M25        (Tool Change)

18   TOOL CALL       2    S600
19   L    Z.1        M03        (M03 turns spindle on clockwise)
20   L    X6,Y3      R0         F2000
21   CYCL DEF  4.1   Set Up -.1
22   CYCL DEF  4.2   Depth      -.25
24   CYCL DEF  4.3   Pecking    _.25       F50
25   CYCL DEF  4.4   X4
26   CYCL DEF  4.5   Y2
27   CYCL DEF  4.6         F40  DR+
```

(These 6 lines answer all of the questions the control needs to ask so that it may execute a canned pocket milling cycle. Refer to the pages following this program for more examples. We have defined a retract plane of .1, it must be given as a negative value, a total depth, a peck depth, the width and length of the pocket, we must be at the pocket's center before executing this cycle, and specified a direction of + about the Cartesian Coordinate Grid. See page 65.)

```
28 CYCL Call   (This tells the control to execute the cycle.)
29   STOP      M25
30   TOOL CALL       3    S1000
31   L    Z.1        R0         F2000
32   L    X.5 Y5.5        (This calls up the drill and moves us to
                          the first hole to be drilled.)
33   CYCL DEF  1.0        Pecking
34   CYCL DEF  1.1        Setup      -.1
35   CYCL DEF  1.2        Depth      -1.25
36   CYCL DEF  1.3        Peckg      -.4
37   CYCL DEF  1.4        Dwell      1 (Dwell time in seconds.)
38   CYCL DEF  1.5        F100
```

(These lines define the canned drilling cycle. See page 67.)

```
39   CYCL CALL            M89        (M89 causes the canned cycle
                                      to remain modal.)
40   L    X 11.5
41   CYCL CALL            M89        (We move to the next hole
                                      location and call the cycle
                                      again.)
42   L    Y.5
43   CYCL CALL            M89
44   L    X.5
45   CYCL CALL            M99        (M99 cancels the canned
                                      cycle.)
46   STOP                 M25
47   TOOL CALL       2    S1000
48   L    X6,Y3      R0   F2000
49   L    Z.1
```

```
50   L     Z-.75        R0    F50
51   L     X6.5,Y3      RL
52   RND         R.1           (This puts a corner round of a specified
                                size.  You must move to the point in the
                                program and then call up the corner
                                round.  I did this here and at the end
                                of my circular path to eliminate any
                                "scallops".  See page 66.)

53   CC    X6,Y3                (This defines the circle center for any
                                succeeding circular moves.  See page
                                62.)

54   C     X6.5,Y3              DR+          F100      (This describes the
                                                      endpoints of the
                                                      circular move.  See
                                                      page 62.)

55   RND         R.1
56   L     X6, Y3       R0
57   L     Z.1          F2000
58   END PGM    1247           End
```

One of the very helpful features of the Heidenhain is that it will allow you to view a graphic simulation of your program. This is done via the following steps.

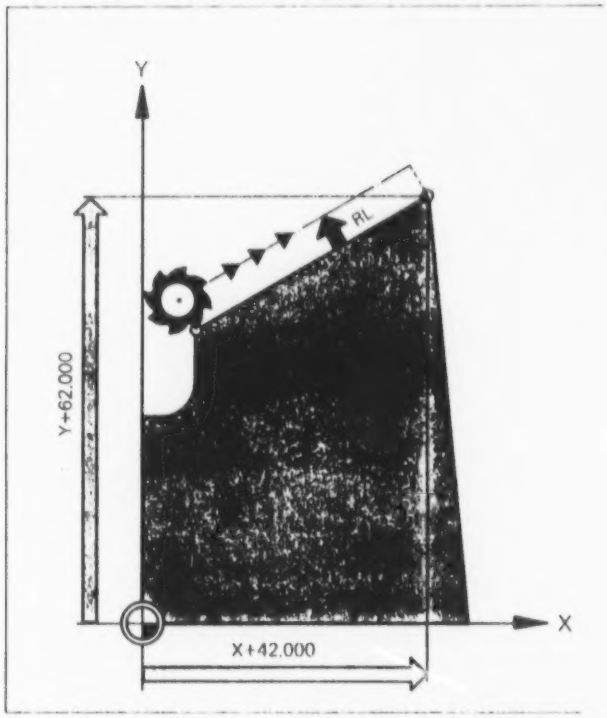
1. Key the RUN button.
2. Enter the program number.
3. Key the BLK FORM button.
4. Key the GRAPHICS button.
5. Pick Plan View 1.
6. Key the START button.

On the following pages are more metric examples of the proper line entries for a variety of situations. Study them carefully.

Linear interpolation

Cartesian (right-angled) co-ordinates

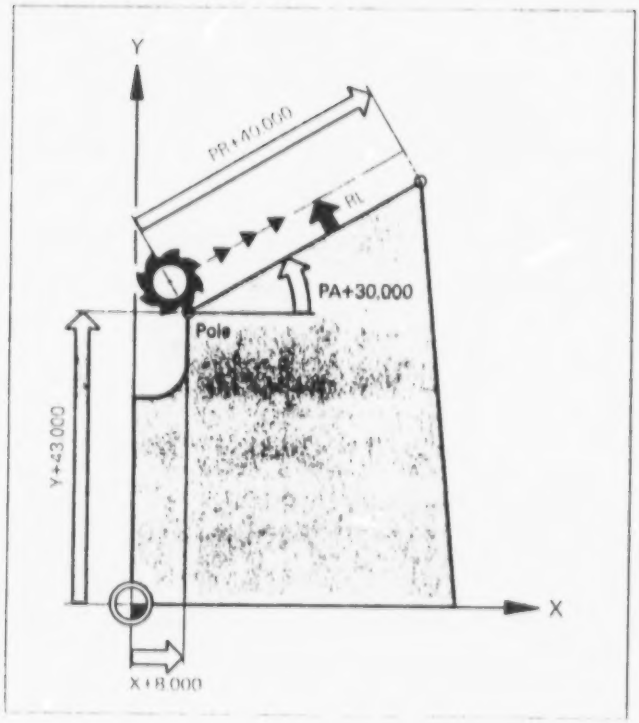
49 L X+42,000 Y+62,000
RL F180 M



Linear interpolation

Polar co-ordinates

```
57 CC X+8,000    Y+43,000
58 LP PR+40,000  PA+30,000
                 RL  F  M
```



Before entry of polar co ordinates, define pole via the CC key

The pole must be programmed in Cartesian co ordinates only!

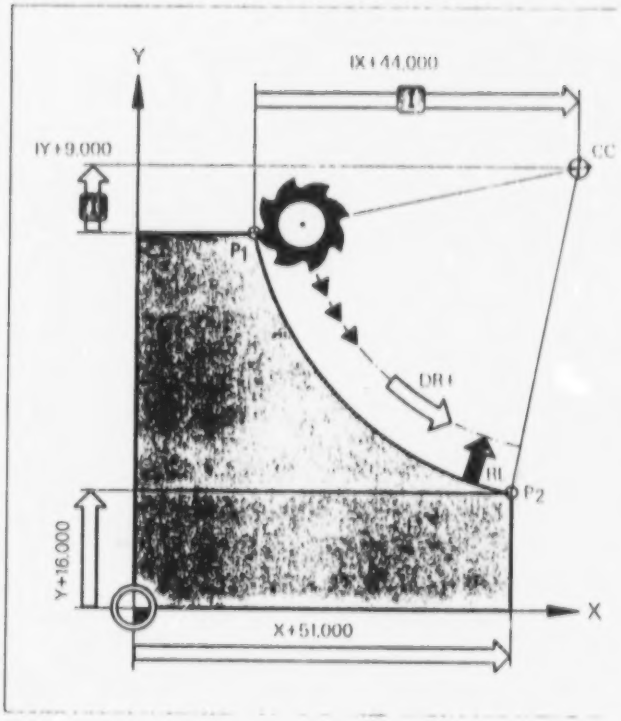
If the pole remains the same for several blocks, a new entry is not required

Circular interpolation

Specification via center and end position
End position in Cartesian (right-angled) co-ordinates

Circle center — 5 CC IX+44,000 IY+9,000
Contour position P_2 — 6 C X+51,000 Y+16,000
DR+ RL F120 M

Contour position P_1 (X16/Y51) has been previously approached



- CC in absolute dimension is referenced to the program zero point

- CC in incremental dimension is referenced to the previous contour position

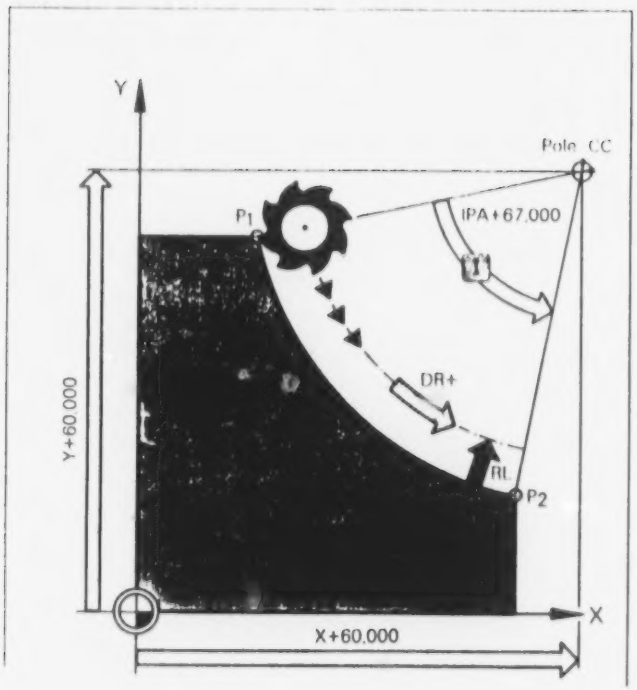
BEST COPY AVAILABLE

Circular interpolation

Specification via center and end position
End position in polar co-ordinates

```
5 CC X+60,000   Y+60,000
6 CP IPA+67,000
        DR+ RL F120 M
```

Circle center/pole
Contour position P_2



Before entry of polar co-ordinates, define pole via the CC-key.

The pole must be programmed in Cartesian co-ordinates only!

Determination of the arc end position only with polar co-ordinate angle PA

CYCL DEF

Slot milling

```
14 CYCL  DEF  3.0   SLOT MILLING
15 CYCL  DEF  3.1   SET-UP -6,000
16 CYCL  DEF  3.2   DEPTH -13,000
17 CYCL  DEF  3.3   PECKG -6,500
    F60
18 CYCL  DEF  3.4   X -30,500
19 CYCL  DEF  3.5   Y +10,500
20 CYCL  DEF  3.6   F180
```

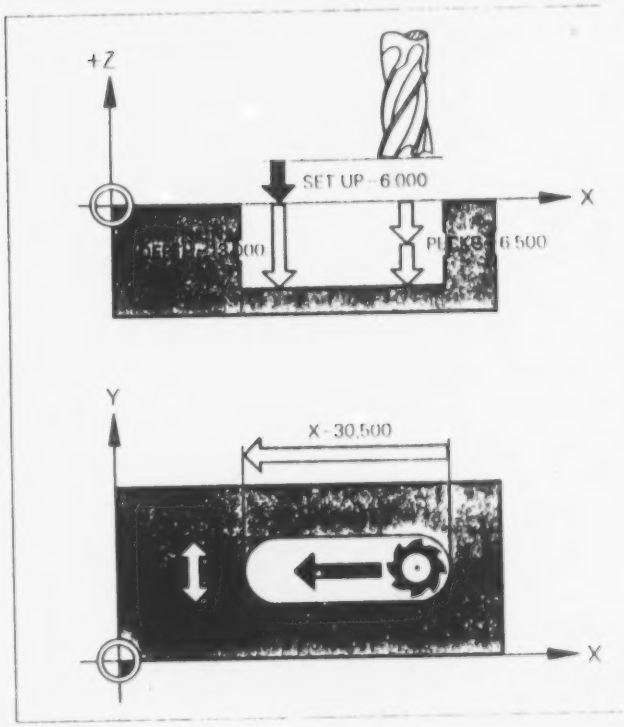
Slot length

Slot width

Arithmetical signs of cycle parameters are entered in accordance with the direction of tool movement.

The sign for the first side length depends on the direction of tool movement
Positive axis direction +
Negative axis direction –

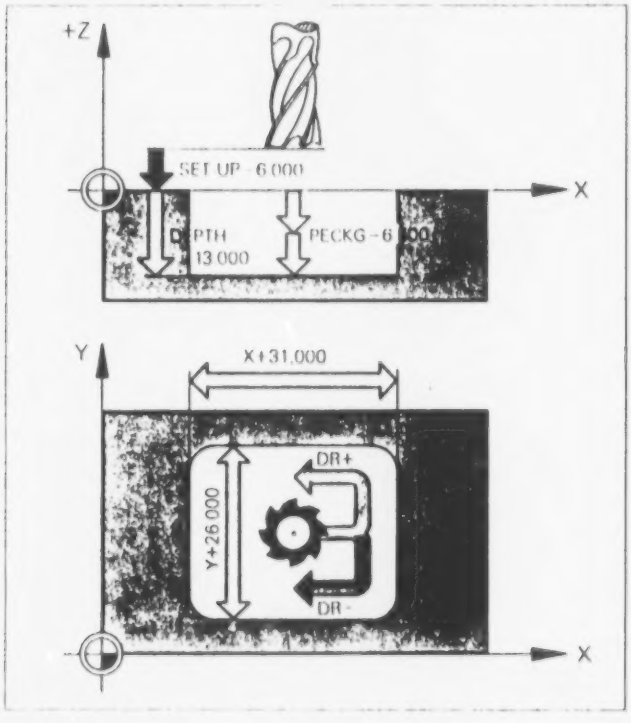
The sign for the second side length is always positive



Rectangular pocket milling

CYCL DEF

```
45 CYCL DEF  4.0  POCKET MILLING
46 CYCL DEF  4.1  SET-UP -6,000
47 CYCL DEF  4.2  DEPTH -13,000
48 CYCL DEF  4.3  PECKG -6,500
     F40
49 CYCL DEF  4.4  X+31,000      First side length
50 CYCL DEF  4.5  Y+26,000      Second side length
51 CYCL DEF  4.6    F140 DR+
```



Arithmetical signs of cycle parameters are entered in accordance with the direction of tool movement

Signs for both side lengths are always positive

DR+ down cut milling

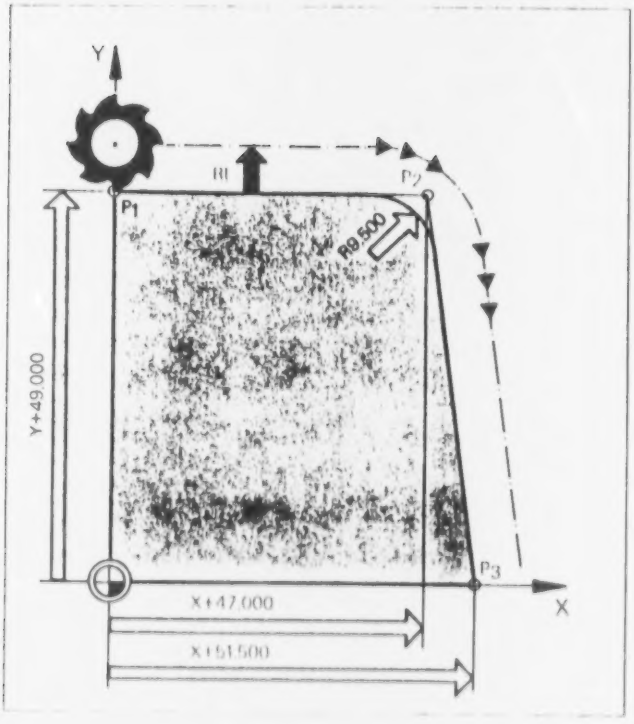
DR - up cut milling

Starting direction positive axis direction of longer side

BEST COPY AVAILABLE

Rounding of corners

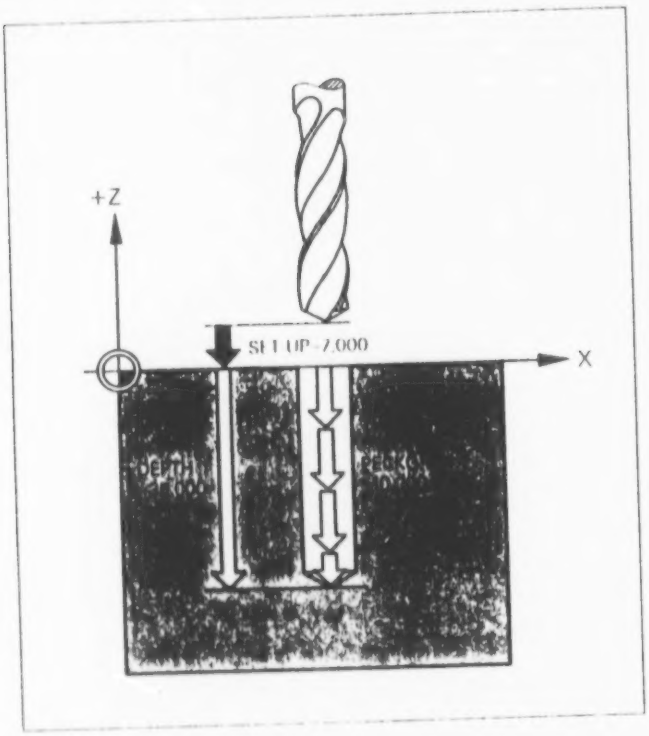
39	L	X+47,000	Y+49,000 RL F120 M	Auxiliary position P_2
40	RND	R9,500	F80	Rounding off radius R
41	L	X+51,500	Y+0,000 RF120 M	Contour position P_3



CYCL DEF

Peck-drilling

```
12 CYCL DEF 1.0 PECKING
13 CYCL DEF 1.1 SET-UP -7,000
14 CYCL DEF 1.2 DEPTH -35,000
15 CYCL DEF 1.3 PECKG -10,000
16 CYCL DEF 1.4 DWELL 1,000
17 CYCL DEF 1.5 F180
```

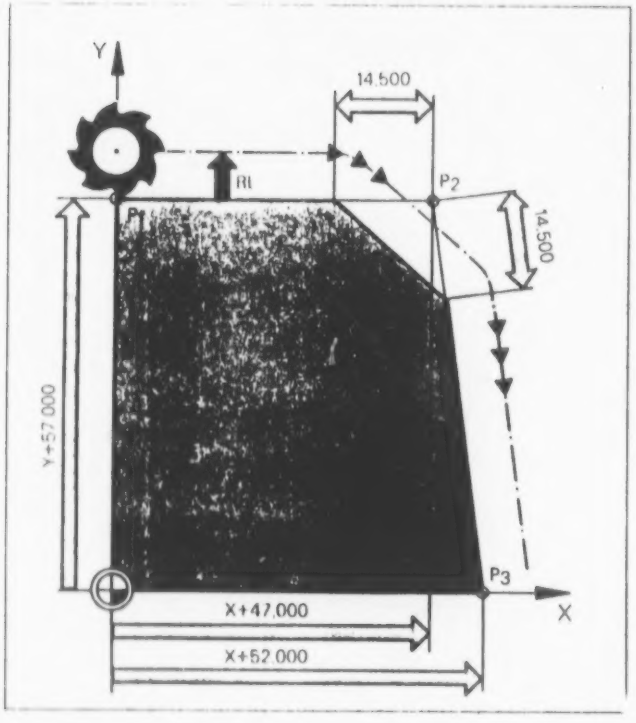


Arithmetical signs of cycle parameters are entered in accordance with the direction of tool movement

BEST COPY AVAILABLE

Chamfer

Auxiliary position P2	95	L X+47,000	Y+57,000 RL F180 M
Chamfer side length L	96	L 14,500	
Contour position P3	97	L X+52,000	Y+0,000 R F M

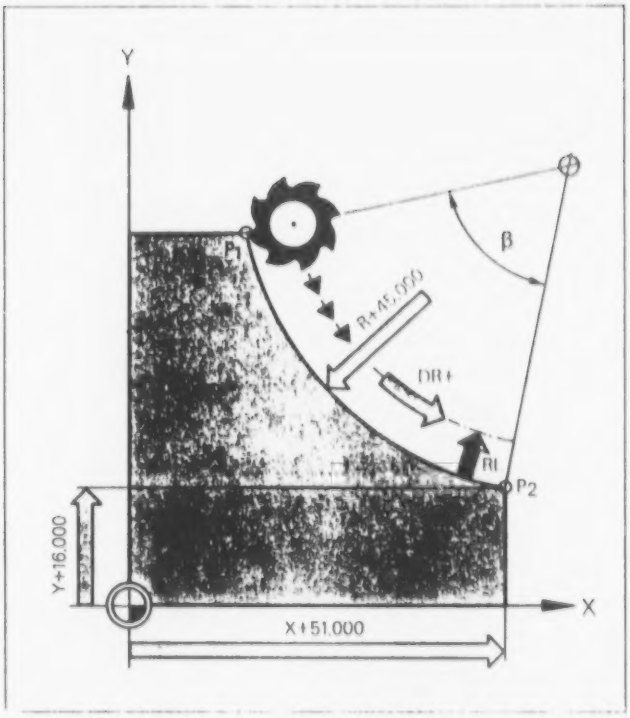


Circular interpolation

Specification via radius and end position

Contour position P_2

```
6  CR X+51,000   Y+16,000
   R+45,000      DR+RL F120 M
```



Sign R

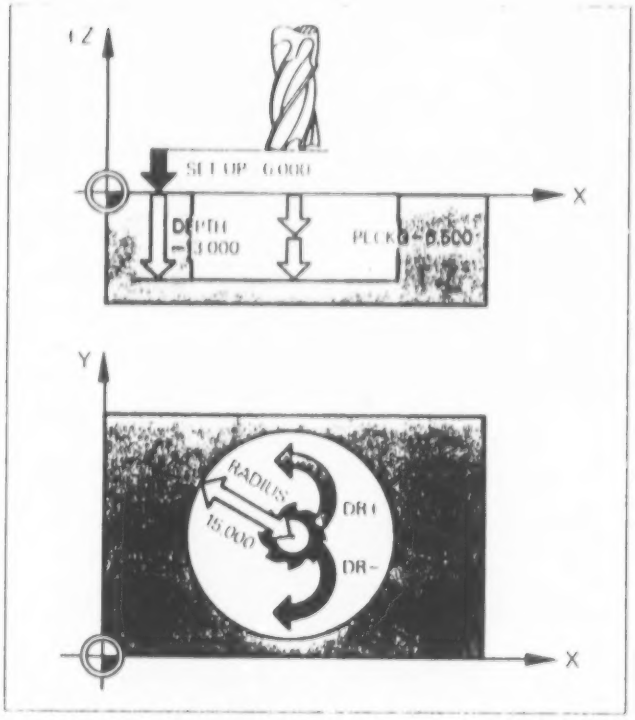
Center square β less than 180°: R+

Center square β greater than 180° R–

CYCL DEF

Circular pocket milling

```
44 CYCL DEF 5.0 CIRCULAR POCKET
45 CYCL DEF 5.1 SET-UP -6,000
46 CYCL DEF 5.2 DEPTH -13,000
47 CYCL DEF 5.3 PECKG -6,500
     F60
48 CYCL DEF 5.4 RADIUS 15,000
49 CYCL DEF 5.5 F120 DR-
```



Arithmetical signs of cycle parameters are entered in accordance with the direction of tool movement

DR+ down cut milling

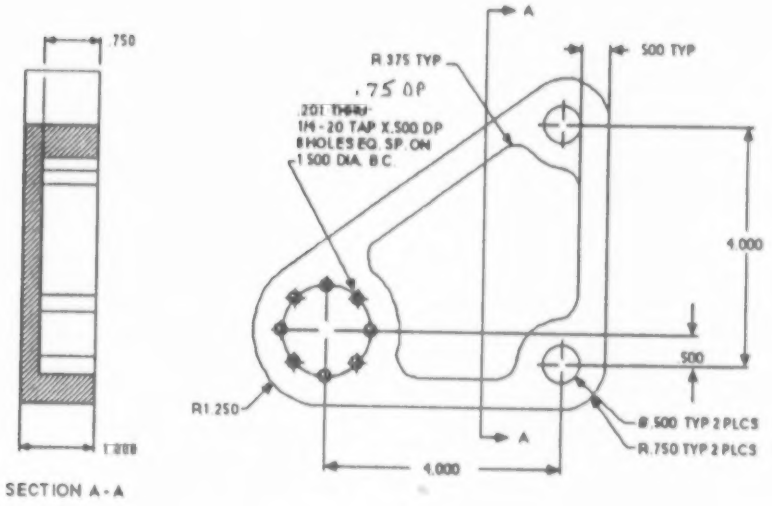
DR- up cut milling

MILWAUKEE AREA TECHNICAL COLLEGE

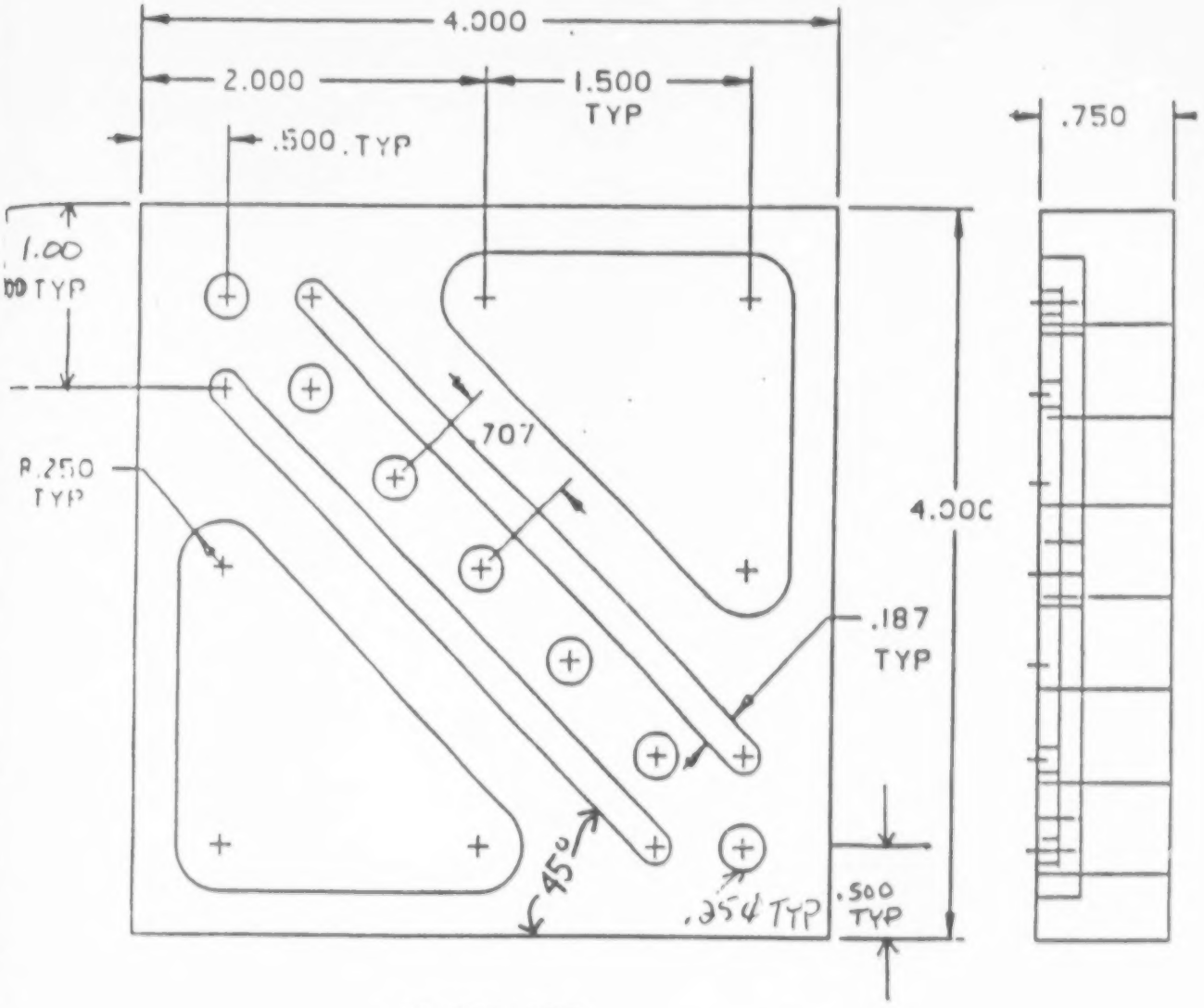
CNC PROGRAMMING I

ASSIGNMENT UNIT 5

1. Write a program which will produce the following part, using the Dialog method for the Heidenhain control. Machine the part using the Heidenhain controlled milling machine. Consult with the instructor regarding speeds, feeds, tools, and Z depths.



2. Write a program which will produce the following part, using the Dialog method for the Heidenhain control. Machine the part using the Heidenhain controlled milling machine. Consult with the instructor regarding speeds, feeds, tools, and Z depths.



SLOTS ARE
.125 DEEP

POCKETS ARE
.250 DEEP

APPENDIX

CNC TOOL LIST AND OPERATION

Part Name	Part No.	Tool Library	Machine No.

Tool Seq. No.	Tool No.	Tool Description and Operation	Tool Dia.	Tool Offset No.	R.P.M.	Feed I.P.M.

PREPARED BY	DATE
CHECKED BY	DATE
SHEET OF	TAPE NO

PART NO.
PART NAME
COMPANY NAME

NUMERICAL CONTROL PROGRAM

NOTES	N	G	X/Y	X/Y/Z	Y/Z/I/J	Y/Z/J/K	F	S	T	M